

**OTOMATISASI PENENTUAN HARGA BEDASARKAN BERAT DAN
VOLUME BARANG PADA JASA PENGIRIMAN
(BAGIAN II)**

TUGAS AKHIR



Oleh :

NUR DEWI RATNA SWARI

NIM. 081310213040

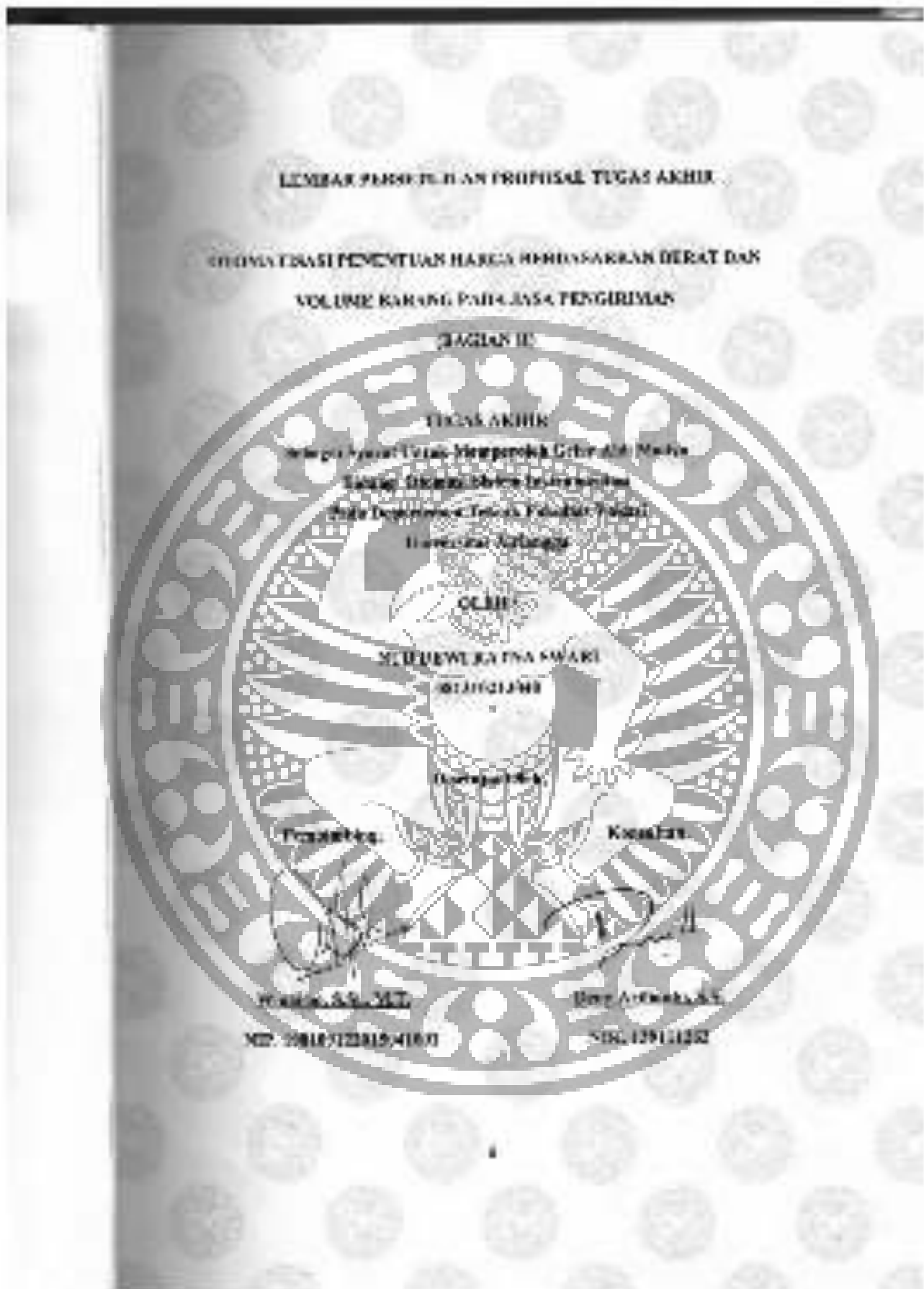
PROGRAM STUDI D3 OTOMASI SISTEM INSTRUMENTASI

DEPARTEMEN TEKNIK

FAKULTAS VOKASI

UNIVERSITAS AIRLANGGA

2016



LEMBAR PENGESAHAN PROPOSAL TUGAS AKHIR

Judul : Observasi Persebaran Rangsang Berakurasi Besar dan Polarisasi Pada Jala Pengiraman

Penyusun : Nur Dewi Ratna Swati

NIM : 001510112640

Tanggal Ujian : 1 Agustus 2015

Pembimbing : Murniati, S.Si., M.T.

Dosen Pembimbing : Dwi Anindya, S.Si.

Dwi Anindya, S.Si.

Dwi Anindya, S.Si.

Widiana, S.Si., M.T.

Dwi Anindya, S.Si.

NIP. 196706120019041001

NIP. 19810103

Murniati, S.Si., M.T.

Ketua Departemen Tata Fisika

Koordinator Program Studi
D3 Teknik Fisika
Instrumen

Dr. Hery Heryanto, S.Si.

Widiana, S.Si., M.T.

NIP. 19671111199041001

NIP. 198109121819043001

KATA PENGANTAR

Segala Puji syukur penulis panjatkan atas kehadiran Allah SWT yang memberikan segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan Proposal Proyek Akhir yang berjudul “Otomatisasi Penentuan Harga Berdasarkan Berat dan Volume Pada Jasa Pengiriman”.

Selama menyusun proposal proyek akhir ini, banyak bantuan moril maupun materil yang telah penulis peroleh dari berbagai pihak, baik secara langsung maupun tidak langsung. Untuk itu dengan segala kerendahan hati, penulis menyampaikan terima kasih kepada :

1. Allah SWT yang telah memberikan ridho, hidayah dan anugerah yang luar biasa.
2. Keluarga tercinta, Ayah, Ibu, Kakak dan Adik tersayang yang telah memberikan segenap do'a dan dukungan kepada penulis.
3. Bapak Winarno, M.Si., selaku Ketua Program Studi D3 Otomasi Sistem Instrumentasi Departemen Teknik Universitas Airlangga Surabaya.
4. Bapak Winarno, selaku Dosen Pembimbing yang telah banyak memberikan arahan, bimbingan, masukan, kepada penulis sehingga terselesaikannya Proyek Akhir ini.
5. Bapak Deny Arifianto, selaku Konsultan yang banyak memberikan arahan, bimbingan, masukan, beserta ketulusan hati dalam membimbing penulis hingga terselesaikannya Proyek Akhir ini.

6. Bapak Tri Anggono, selaku Tim Quality Control yang telah menyetujui judul Tugas Akhir ini.
7. Rumpita OSI'13 Wanda, Sofi, Ifa, Media, Dita, Aliyah, Oneng, dan Miming yang selalu memberi semangat dan mendukung setiap keputusan yang terbaik untuk Tugas Akhir ini.
8. Renaldy Mulya sebagai partner Tugas Akhir yang membantu menyelesaikan Tugas Akhir ini.
9. Farieddudin sebagai teman yang ikut serta meminjamkan mouse untuk kelancaran Tugas Akhir ini.
10. Rekan-rekan seperjuangan OSI'2013, kalian penyemangat kami yang tak henti-hentinya mengingatkan dan menegur atas segala kekhilafan kami. Semoga tahun ini kita lulus bersama-sama. Amin
11. Semua pihak yang telah membantu yang tidak bisa disebutkan satu per satu, kami ucapkan terimakasih banyak.

Penulis menyadari masih banyak kekurangan dalam penulisan proposal ini, oleh karena itu kritik dan saran yang sifatnya membangun sangat diharapkan demi kesempurnaan laporan ini. Namun penulis berharap proposal ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan perkembangan teknologi.

Surabaya, 15 Juli 2016

Penulis

Nur Dewi Ratna Swari, 2016, *Sistem Otomatis Penentuan Harga Berdasarkan Berat dan Volume Barang Pada Jasa Pengiriman (Bagian II)*. Tugas Akhir ini di bawah bimbingan Winarno, S.Si., M.T. dan Deny Arifianto S.Si. Prodi D3 Otomasi Sistem Instrumentasi Departemen Teknik Fakultas Vokasi Universitas Airlangga.

ABSTRAK

Sistem kerja dengan cara pengoperasian secara otomatis mulai banyak digunakan. Pada bidang industri, pengoperasian secara otomatis sangatlah berguna untuk menyelesaikan serta mempermudah pekerjaan. Tapi tidak hanya pada bidang industri, pada bidang jasa atau pelayanan juga memerlukan pengoperasian yang otomatis juga dan itu sangat bermanfaat. Berdasarkan hal tersebut, pada tugas akhir ini merancang sebuah sistem otomatis yang dapat memudahkan dalam pengukuran dimensi dan berat benda pada jasa pengiriman dan menghasilkan data yang akurat. Rancang bangun alat ini menggunakan sensor SRF-05 sebagai pengukur dimensi benda, *strain gauge* sebagai pengukur massa benda, Arduino UNO, dan LCD sebagai tampilan. Apabila telah didapat data dari panjang, lebar, tinggi dan berat barang maka data tersebut akan dikirimkan ke *Labview* yang kemudian akan ditampilkan harganya. Dalam melakukan percobaan pengukuran serta penimbangan sebanyak 8 kali, pengujian keseluruhan sistem mengalami simpangan rata-rata ± 0.2 untuk pengujian perbandingan berat terukur dan berat asli dan ± 0.18 untuk pengujian perbandingan volume terukur dan volume asli. Dalam hal ini sistem memiliki nilai simpangan yang kecil sehingga keseluruhan sistem dapat bekerja dengan baik.

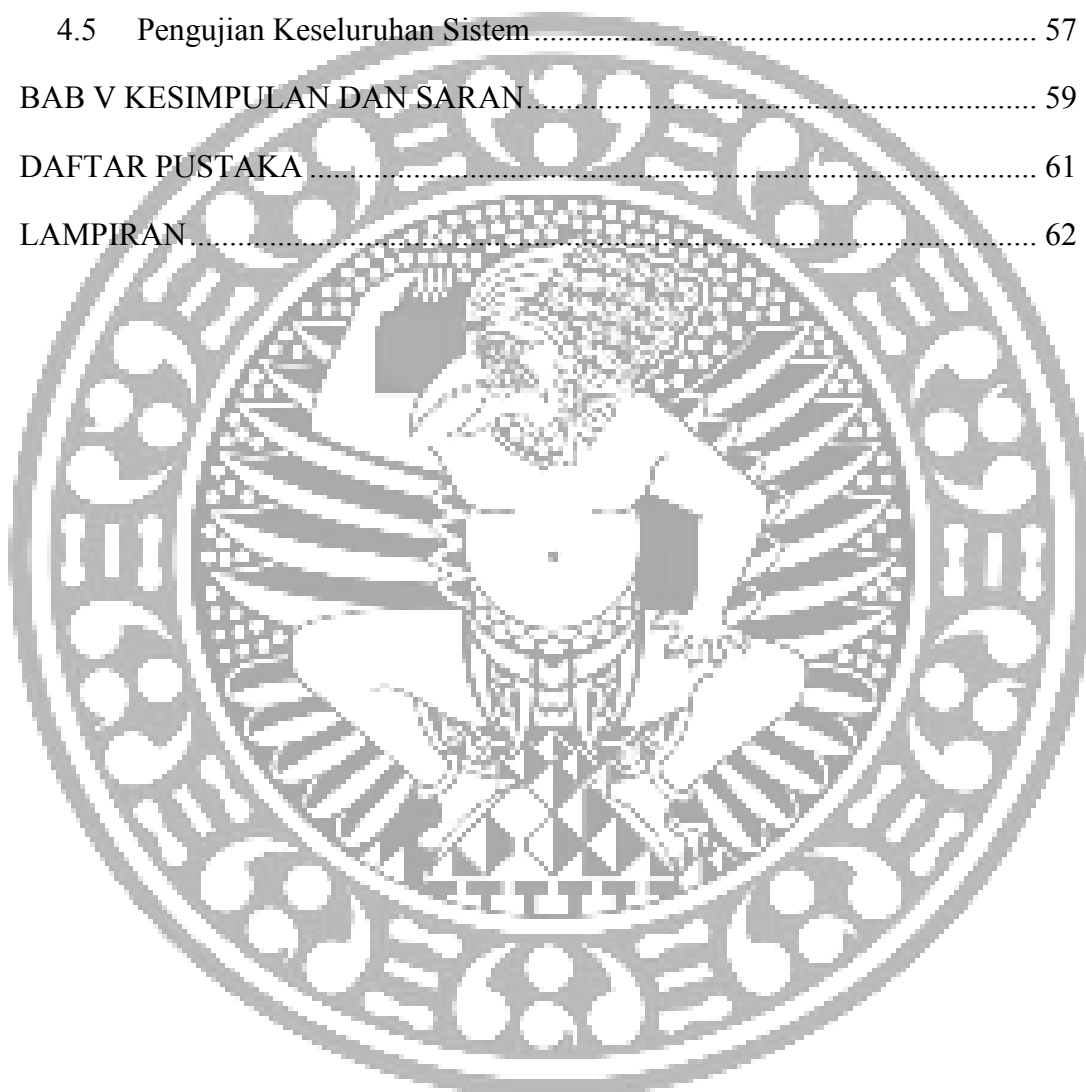
Kata Kunci : Mikrokontroler, SRF05, *Strain gauge*, HX711, *Labview*.

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat Tugas Akhir.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Jasa Pengiriman	4
2.2 Pengenalan LabView.....	6
2.3 Pemrograman Bahasa C.....	9
2.3.1 Bahasa Pemrograman Arduino.....	9
2.3.2 Dasar- dasar Pemrograman C.....	12
2.3.3 <i>Flow Control</i>	13

2.4	IDE Arduino	14
2.5	HX711	16
2.6	Sensor SRF-05.....	17
2.7	LCD 16x2	18
2.4	Mekanisme Kerja Alat.....	19
BAB III METODE PENELITIAN.....		20
3.1	Waktu dan Tempat Pelaksanaan Penelitian.....	20
3.2	Alat dan Bahan	20
3.2.1	Alat.....	20
3.2.2	Bahan.....	20
3.3	Prosedur Penelitian.....	21
3.3.1	Tahap Persiapan Alat	22
3.3.2	Tahap Perancangan Alat	22
3.3.3	Tahap Perwujudan Alat.....	22
3.3.4	Tahap Pengujian Alat.....	30
3.3.5	Analisis Data	38
BAB IV HASIL DAN PEMBAHASAN.....		40
4.1	Hasil Pembuatan Perangkat Lunak (Software).....	40
4.1.1	Sub Program Awal	40
4.1.2	Sub Pengaturan Awal Kerja Program	42
4.1.3	Sub Program Data	43
4.1.4	Sub Program Rerata	44
4.1.5	Sub Program Penentuan Berat	45
4.1.6	Sub Program Reset dan Tampil Pada LCD.....	46
4.1.7	Sub Program Perulangan.....	46

4.1.8	Interface Sistem Menggunakan Labview.....	48
4.2	Pengujian Sensor SRF-05.....	49
4.3	Pengujian Sensor <i>Strain Gauge</i>	55
4.4	Pengujian Lama Waktu Menentukan Berat Barang Berdasarkan Berat atau Volume	57
4.5	Pengujian Keseluruhan Sistem.....	57
BAB V KESIMPULAN DAN SARAN.....		59
DAFTAR PUSTAKA.....		61
LAMPIRAN.....		62



DAFTAR TABEL

Tabel 2.1 Simbol Aritmatika.....	13
Tabel 3.1 Pengujian Jarak Mistar Terhadap Waktu Tempuh Sensor SRF-05 A	31
Tabel 3.2 Pengujian Jarak Mistar Terhadap Waktu Tempuh Sensor SRF-05 B.....	31
Tabel 3.3 Pengujian Jarak Mistar Terhadap Waktu Tempuh Sensor SRF-05 C.....	32
Tabel 3.4 Linieritas Jarak Mistar Terhadap SRF-05 A Bahan Kardus	33
Tabel 3.5 Linieritas Jarak Mistar Terhadap SRF-05 B Bahan Kardus	33
Tabel 3.6 Linieritas Jarak Mistar Terhadap SRF-05 C Bahan Kardus	34
Tabel 3.7 Linieritas Jarak Mistar Terhadap SRF-05 A Bahan Plastik.....	35
Tabel 3.8 Linieritas Jarak Mistar Terhadap SRF-05 B Bahan Plastik.....	36
Tabel 3.9 Linieritas Jarak Mistar Terhadap SRF-05 C Bahan Plastik.....	36
Tabel 3.10 Pengujian Linieritas Sensor <i>Strain Gauge</i>	37
Tabel 3.11 Tabel Lama waktu Mengukur dan Menimbang Barang	39
Tabel 3.12 Pengujian Keseluruhan Sistem.....	39

DAFTAR GAMBAR

Gambar 2.1 <i>Box</i> Barang	5
Gambar 2.2 Contoh Front Panel Pada Sistem	7
Gambar 2.3 Contoh Blog Diagram Vi Sistem	7
Gambar 2.4 <i>Contoh Control dan Function Pallete</i>	8
Gambar 2.5 Program Pada IDE Arduino	15
Gambar 2.6 Skema HX711	16
Gambar 2.7 Grafik Proses Konversi HX711	16
Gambar 2.8 Diagram Timing SRF-05 Mode I dan Mode II	18
Gambar 2.9 Konfigurasi pin LCD 16x2 dan Modul I2C	19
Gambar 3.1 Diagram Prosedur Kerja	21
Gambar 3.2 Desain Mekanik Alat	24
Gambar 3.3 Desain Mekanik Tampak Bawah dan Atas	24
Gambar 3.4 Desain Mekanik Tampak Belakang	25
Gambar 3.5 Desain Mekanik Tampak Kanan dan Kiri	25
Gambar 3.6 Blok Diagram Alat	26
Gambar 3.7 Diagram <i>Flow Control</i>	27
Gambar 3.8 Diagram Alir Pada Arduino UNO	28
Gambar 3.9 Diagram Alir Pada LabView	29
Gambar 4.1 Gambar Tampilan <i>Interface</i> Sistem Penentuan Harga	48
Gambar 4.2 Grafik Fungsi Jarak Terhadap Waktu Tempuh Sensor A	50
Gambar 4.3 Grafik Fungsi Jarak Terhadap Waktu Tempuh Sensor B	50
Gambar 4.4 Grafik Fungsi Jarak Terhadap Waktu Tempuh Sensor C	51
Gambar 4.5 Grafik Hubungan Antara Kalibrator Jarak Mistar Terhadap Sensor SRF-05 A	53
Gambar 4.6 Grafik Hubungan Antara Kalibrator Jarak Mistar Terhadap Sensor SRF-05 B	53

Gambar 4.7 Grafik Hubungan Antara Kalibrator Jarak Mistar Terhadap Sensor SRF-05 C.....	54
Gambar 4.8 Grafik Linieritas Sensor <i>Strain Gauge</i> Dengan Timbangan Digital	56



DAFTAR LAMPIRAN

Lampiran 1 Gambar Sistem Alat.....	62
Lampiran 2 Program Arduino	63
Lampiran 3 Data Sheet HX711	65
Lampiran 4 Data Sheet SRF-05	74



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengiriman jasa dapat mengirimkan segala jenis barang, mulai dari barang berat, ringan, dan juga barang dalam ukuran besar. Hal ini sangat bermanfaat bagi seluruh masyarakat untuk mengirimkan barang. Barang-barang yang dikirimkan oleh masyarakat memiliki berat yang berbeda. Berat barang dibagi menjadi 2 jenis yaitu berat aktual dan berat volume. Berat aktual yang merupakan berat asli dari barang tersebut dan biasanya memiliki berat yang lebih berat dari ukurannya. Sedangkan berat volume merupakan berat yang memiliki ukuran volume yang lebih besar dari nilai berat aslinya. Makin besar dan berat barang yang akan dikirim, juga dapat menentukan harga pengiriman barang tersebut. Harga barang dengan berat volume biasanya digunakan bila mana barang memiliki nilai volume yang lebih besar dari berat aktual. Hal ini diterapkan karena jika diterapkan penentuan harga berdasarkan berat aktual, maka pihak jasa pengiriman akan merasa dirugikan. Sebab dengan ukuran paket yang besar akan memakan tempat cukup banyak dan resikonya lebih besar. Dalam kasus ini, pihak jasa pengiriman akan menerapkan tarif berdasarkan volume.

Pada saat ini, untuk mengetahui volume barang dan berat asli barang harus menggunakan cara manual yaitu manusia menggunakan timbangan analog untuk mendapatkan berat asli barang dan penggaris agar dapat memperoleh nilai panjang, lebar, tinggi dari barang tersebut. Data dan perhitungan berat volume

yang di berikan pada data base jasa pengiriman saat ini juga dilakukan secara manual dan masih menggunakan kalkulator untuk mendapatkan hasil berat volume tersebut. Selain itu, harus membutuhkan waktu yang lama bila harus mengitung dan membandingkan secara manual antara berat volume dan berat aktual agar dapat ditentukan harga tiap kotanya. Dengan menggunakan cara manual tersebut, banyak operator yang membandingkan pengukuran dari barang tersebut menggunakan *felling* mereka untuk mengambil berat pada barang. Hal ini dapat menimbulkan kesalahan yang diakibatkan karena kurang telitinya operator. Hal ini tidak dapat dibiarkan begitu saja karena dapat menimbulkan kerugian baik besar maupun kecil.

Berdasarkan latar belakang yang telah dijelaskan sebelumnya akan dibuat inovasi dalam proses penentuan harga yaitu pembuatan miniature “Otomatisasi Penentuan Harga Berdasarkan Berat Dan Volume pada Jasa Pengiriman”. Pada sistem ini dapat menggunakan sensor *Strain Gauge* dan sensor ultrasonik untuk mengetahui berat dan volume pada barang. Selain itu, sistem ini menjadikan pekerjaan menjadi lebih mudah untuk usaha jasa pengiriman. Dengan hanya meletakkan barang pada sistem ini maka, nilai berat, panjang, lebar, tinggi, dan perhitungan volume akan secara otomatis didapat untuk mengetahui apakah barang termasuk jenis berat aktual atau jenis berat volume tanpa harus membuang waktu. Setelah itu, proses penentuan harga berat barang pada tiap kota akan segera didapatkan untuk di tampilkan pada LCD.

1.2 Rumusan Masalah

Untuk mempermudah mewujudkan sistem yang sesuai dengan tujuan penelitian, disusun rumusan masalah yang mencakup :

1. Bagaimana metode untuk mengukur berat dan volume barang pada proses penentuan harga pada tugas akhir ini?
2. Bagaimana kinerja alat untuk proses penentuan harga pada barang?

1.3 Batasan Masalah

Agar masalah tidak meluas dan menyimpang dari tujuan maka perlu dilakukan pendekatan terhadap yaitu :

1. Sistem yang dibuat berdimensi 60cmx60cmx60cm
2. Otomatisasi penentuan harga pada barang ini hanya digunakan pada barang dengan bentuk *box* yang memiliki berat dan dimensi maksimal sebesar 50cmx50cmx50cm dan 10kg

1.4 Tujuan

Adapun tujuan yang ingin dicapai dalam pembuatan tugas akhir ini adalah:

1. Untuk mengetahui metode untuk mengukur berat dan volume barang pada proses penentuan harga pada tugas akhir ini.
2. Untuk mengetahui kinerja alat untuk proses penentuan harga pada barang.

1.5 Manfaat

Manfaat tugas akhir ini adalah untuk memudahkan operator pengiriman jasa agar dapat menentukan barang dengan jenis berat aktual dan volume dalam proses penentuan harga.

BAB II

TINJAUAN PUSTAKA

2.1 Jasa Pengiriman

Secara umum pelayanan jasa pengiriman barang adalah segala upaya yang diselenggarakan atau dilaksanakan secara sendiri atau secara bersama – sama dalam suatu organisasi untuk memberikan pelayanan secara efektif dan efisien berupa pengiriman barang, baik antar kota, antar pulau dan antar negara. Jenis barang yang dapat dikirimkan dapat berupa dokumen, uang dan barang-barang yang memiliki berat yang besar. Sebagian besar bisnis *online* menggunakan jasa pengiriman sebagai sarana transportasi yang dapat digunakan sebagai pengantar barang pesanan.

Keamanan dalam menjaga barang yang akan dikirimkan agar sampai ketempat tujuan juga sangat dijaga oleh pelaku bisnis pengiriman jasa ini agar konsumen dapat selalu percaya jika ingin mengirimkan suatu barang yang dinilai sangat penting dan berharga. Keamanan ini, berpengaruh pada jumlah biaya pengiriman yang akan konsumen bayarkan.

Penetapan biaya kirim jasa pengiriman dibagi menjadi 2 jenis berat pada setiap barang. Jenis yang pertama adalah penetapan ongkos kirim progresif. Ongkos kirim ini dihitung berdasarkan berat aktual pada barang. Semakin berat barang, ongkos kirim juga semakin besar berdasarkan setiap kotanya. Jenis yang kedua adalah ongkos kirim volumetrik. Ongkos kirim ini berdasarkan berat volume pada barang yang biasanya memiliki ukuran yang lebih besar dan barangnya lebih ringan. Dalam hal ini jika diterapkan ongkos kirim berdasarkan berat asli, maka

pihak jasa pengiriman akan merasa dirugikan. Sebab, dengan ukuran barang yang besar akan memakan tempat cukup banyak dan juga memiliki risiko yang lebih besar. Dalam kasus ini, pihak jasa pengiriman dapat menerapkan tarif berdasarkan volume pada barang dengan menggunakan rumus :

$$M_v = \frac{p \times l \times t}{k} \dots\dots\dots(2.1)$$

Keterangan :

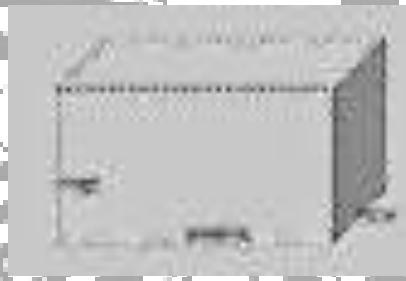
M_v = Berat Volume (Kg)

p = Panjang *box* (cm)

l = Lebar *box* (cm)

t = Tinggi *box* (cm)

k = Konstanta IATA (6000 cm³/kg)



Gambar 2.1 *Box* barang

Dengan menggunakan perhitungan tarif volumetrik, maka perbandingan harga dan besar barang akan berimbang. Perhitungan volumetrik ini menggunakan pembagi 6000cm³/kg yang merupakan konstanta yang ditetapkan oleh *International Air Transport Association* (IATA). Konstanta ini juga digunakan untuk cargo penerbangan udara di Indonesia. Penerbangan udara ini, banyak digunakan sebagai transportasi pengiriman barang pada jasa pengiriman di Indonesia.

Untuk menentukan apakah suatu barang akan dikenakan tarif berdasar berat asli atau tarif volumetrik adalah dilihat mana yang lebih besar ongkos kirimnya. Jika sebuah barang berat asli atau aktualnya hanya 2 kg, sedangkan perhitungan volumetriknya 4 kg, maka berat yang digunakan adalah berat yang lebih besar untuk proses penentuan harga.

2.2 Pengenalan LabVIEW

LabVIEW adalah sebuah *software* pemrograman yang diproduksi oleh National instruments dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau Visual basic, LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa labVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text. Program labVIEW dikenal dengan sebutan Vi atau *Virtual Instruments* karena penampilannya dan operasinya dapat meniru sebuah instrument. Pada labVIEW, *user* pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan *input* lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu *user* menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software* LabVIEW terdiri dari tiga komponen utama, yaitu :

1. Front Panel

Front Panel adalah bagian *window* yang berlatar belakang abu-abu serta mengandung kontrol dan indikator. *Front panel* digunakan untuk

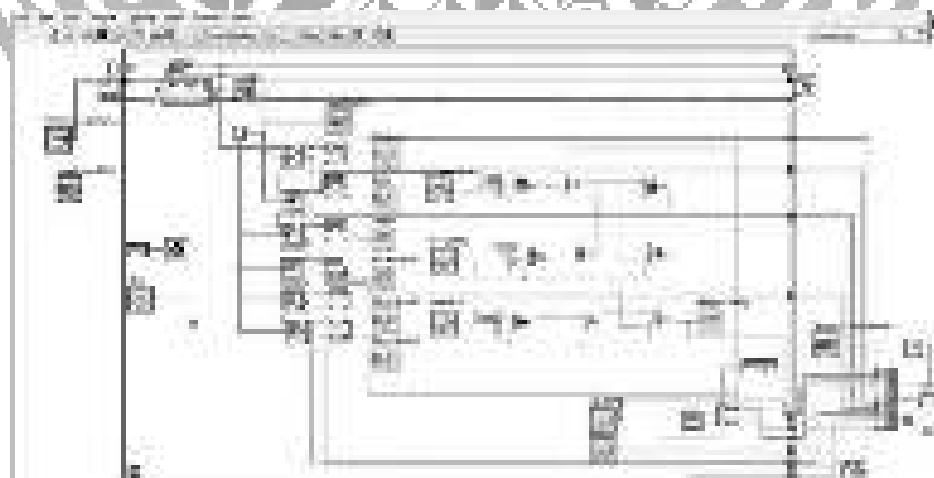
membangun sebuah VI, menjalankan program dan mendebug program. Tampilan dari *front panel* pada sistem dapat di lihat pada Gambar 2.2 ini:



Gambar 2.2 Contoh *Front Panel* Pada Sistem

2. Blok diagram dari Vi

Blok diagram adalah bagian *window* yang berlatar belakang putih berisi source code yang dibuat dan berfungsi sebagai instruksi untuk *front panel*. Tampilan dari blok diagram pada sistem dapat di lihat pada Gambar 2.3 ini:



Gambar 2.3 Contoh Blok Diagram Vi Pada Sistem

3. *Control dan Functions Pallette*

- a. *Control dan Functions Pallette* digunakan untuk membangun sebuah Vi. *Control Pallette* merupakan tempat beberapa *control* dan indikator pada *front panel*, *control pallette* hanya tersedia di *front panel*, untuk menampilkan *control pallette* dapat dilakukan dengan mengklik *windows >> show control pallette* atau klik kanan pada *front panel*. Contoh *control pallette* pada sistem dapat dilihat pada Gambar 2.4 ini:



Gambar 2.4 Contoh *Control Pallette* Pada Sistem

b. *Functions Pallette*

Functions Pallette digunakan untuk membangun sebuah blok diagram, *Functions pallette* hanya tersedia pada blok diagram, untuk menampilkannya dapat dilakukan dengan mengklik

windows >> show control pallete atau klik kanan pada lembar kerja blok diagram.

2.3 Pemrograman Bahasa C

Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada di antara bahasa tingkat rendah dan tingkat tinggi. Bahasa tingkat rendah artinya bahasa yang berorientasi pada mesin dan tingkat tinggi berorientasi pada manusia. Bahasa tingkat rendah, misalnya bahasa assembler, bahasa ini ditulis dengan sandi yang dimengerti oleh mesin saja, oleh karena itu hanya digunakan bagi yang memprogram mikroprosesor. Bahasa tingkat rendah merupakan bahasa yang membutuhkan kecermatan yang teliti bagi pemrogram karena perintahnya harus rinci, ditambah lagi masing-masing pabrik mempunyai sandi perintah sendiri. Bahasa tinggi relatif mudah digunakan, karena ditulis dengan bahasa manusia sehingga mudah dimengerti dan tidak tergantung mesinnya. Bahasa tingkat tinggi biasanya digunakan pada komputer.

2.3.1 Bahasa Pemrograman Arduino

Program yang Arduino gunakan adalah bahasa C. Sebagian besar dari paraprogramer profesional masih tetap memilih bahasa C sebagai bahasa yang lebih unggul. Struktur penulisan program dalam bahasa C harus memiliki fungsi utama, yang bernama `main()`. Fungsi inilah yang akan dipanggil pertama kali pada saat proses eksekusi program. Artinya apabila kita mempunyai fungsi lain selain fungsi utama, maka fungsi lain tersebut baru akan dipanggil pada saat digunakan.

Selain itu juga dalam bahasa C kita akan mengenal *file header*, biasa ditulis dengan ekstensi *h(*.h)*, adalah file bantuan yang digunakan untuk menyimpan daftar-daftar fungsi yang akan digunakan dalam program. Bagi anda yang sebelumnya pernah mempelajari bahasa pascal, *file header* ini serupa dengan unit. Dalam bahasa C, file header standar yang untuk proses *input/output* adalah *<stdio.h>*.

Perlu sekali untuk diperhatikan bahwa apabila kita menggunakan *file header* yang telah disediakan oleh kompilator, maka kita harus menuliskannya didalam tanda '*<*' dan '*>*' (misalnya *<stdio.h>*). Namun apabila menggunakan *file header* yang kita buat sendiri, maka file tersebut ditulis di antara tanda “ dan ” (misalnya “*cobaheader.h*”). perbedaan antara keduanya terletak pada saat pencarian file tersebut. Apabila kita menggunakan tanda *<>*, maka file tersebut dianggap berada pada direktori default yang telah ditentukan oleh kompilator. Sedangkan apabila kita menggunakan tanda “”, maka *file header* dapat kita dapat tentukan sendiri lokasinya. *File header* yang akan kita gunakan harus kita daftarkan dengan menggunakan directive *#include*. Directive *#include* ini berfungsi untuk memberi tahu kepada kompilator bahwa program yang kita buat akan menggunakan file-file yang didaftarkan. Berikut ini contoh penggunaan directive *#include*.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include"myheader.h"
```

Setiap kita akan menggunakan fungsi tertentu yang disimpan dalam sebuah *file header*, maka kita juga harus mendaftarkan *file header*nya dengan menggunakan directive *#include*. Sebagai contoh, kita akan menggunakan fungsi *getch()* dalam program, maka kita harus mendaftarkan *file header* *<conio.h>*.

Berikut adalah contoh penulisan program dalam bahasa C :

1. Structure

Struktur dasar dari dasar pemrograman arduino sangatlah sederhana yaitu terdiri dari 2 bagian, yaitu:

```
void setup()  
{  
  // Statement;  
}  
  
void loop()  
{  
  // Statement;  
}
```

Dimana *setup()* untuk inisialisasi yang hanya dijalankan sekali di awal program, sedangkan *loop()* untuk mengeksekusi bagian program yang akan dijalankan berulang-ulang.

2. Setup ()

Fungsi *setup()* hanya di panggil satu kali ketika program pertama kali di jalankan. Ini digunakan untuk pendefinisian mode pin atau memulai komunikasi serial. Fungsi *setup()* harus di ikut sertakan dalam program walaupun tidak ada statement yang di jalankan.

```
void setup()  
{  
  pinMode(13,OUTPUT); // mengset 'pin' 13 sebagai output  
}
```


3. Loop ()

Setelah melakukan fungsi setup() maka secara langsung akan melakukan fungsi loop() secara berurutan dan melakukan instruksi-instruksi yang ada dalam fungsi loop().

```
void loop()
{
digitalWrite(13, HIGH);    // nyalakan 'pin' 13
delay(1000);               // pause selama 1 detik
digitalWrite(13, LOW);     // matikan 'pin' 13
delay(1000);               // pause selama 1 detik
}
```

2.3.2 Dasar-dasar Pemrograman C

Pada bahasa C, dasar-dasar yang pemrograman perlu diketahui antara lain:

1. Tipe Data Dasar

Data merupakan suatu nilai yang biasa dinyatakan dalam bentuk konstanta atau variabel. Konstanta menyatakan nilai yang tetap, sedangkan variable menyatakan nilai yang dapat diubah-ubah selama eksekusi berlangsung. Tipe data pemrograman bahasa C dapat dibagi menjadi beberapa tipe sebagai berikut:

Char : 1 byte (-128 s/d 127)

Unsigned char : 1 byte (0 s/d 255)

Int: 2 byte (-32768 s/d 32767)

Long: 4 byte (2147483648 s/d -2147483647)

Unsigned long: 4 byte (0 s/d 4294967295)

Array: kumpulan data-data yang sama tipenya.

Operator atau tanda operasi adalah suatu tanda atau simbol yang akan untuk suatu operasi tertentu. Jenis-jenis simbol aritmatika dapat dilihat Tabel 2.1 berikut.

Tabel 2.1 Simbol Aritmatika

Kontrol aliran pada bahasa C dapat digunakan untuk pengkondisian masing-masing program pada perangkat sesuai dengan keinginan. Berikut adalah beberapa pengkondisian yang sering digunakan:

Operator if digunakan untuk sebuah kondisi seperti nilai analog sudah berada di bawah nilai yang kita kehendaki atau belum, apabila terpenuhi maka

akan mengeksekusi baris program yang ada dalam *brackets* kalau tidak terpenuhi maka akan mengabaikan baris program yang ada dalam *brackets*.

If (someVariable ?? value)

```
{  
  //DoSomething;  
}
```

2. if... else

Operator if...else mengtes sebuah kondisi apabila tidak sesuai dengan kondisi yang pertama maka akan mengeksekusi baris program yang ada di else.

```
If ( inputPin == HIGH )  
{  
  //Laksanakan rencana A;  
}  
Else  
{  
  //Laksanakan rencana B;  
}
```

3. for

Operator for digunakan dalam pengulangan.

```
For ( initialization; condition; expression )  
{  
  //doSomethig;  
}
```

4. IDE Arduino

Software Arduino yang akan digunakan adalah driver dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan Arduino. IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

- Editor program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.

- Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah *microcontroller* tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh *microcontroller* adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
- Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan Arduino.

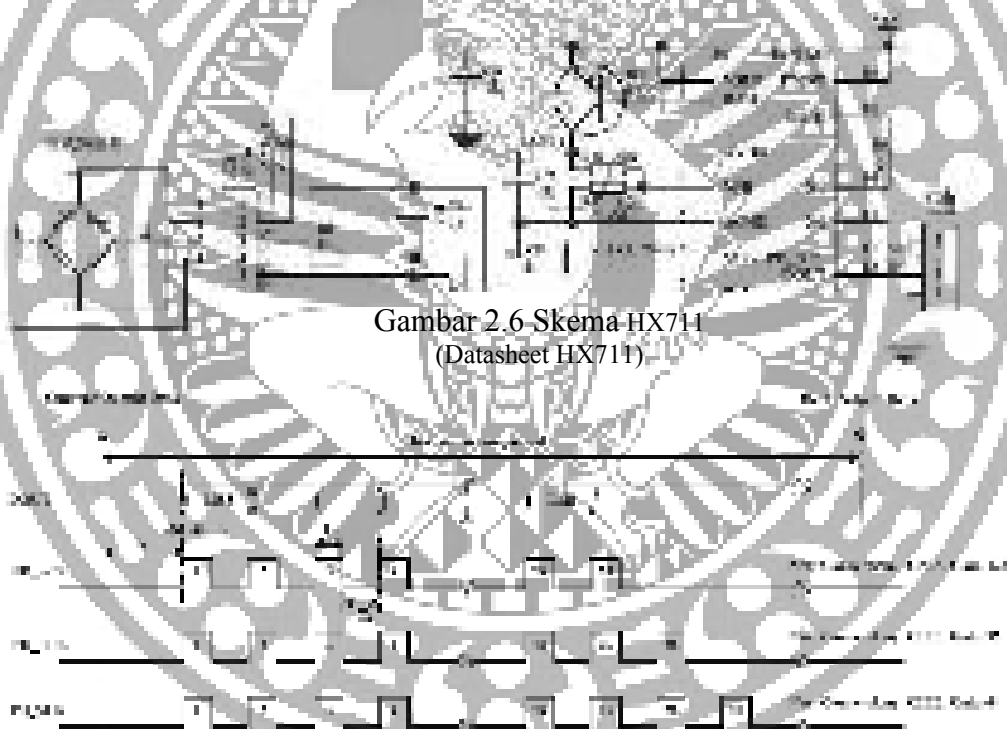
Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata “sketch” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama. Berikut ini adalah contoh tampilan IDE Arduino:



Gambar 2.5 Contoh Program IDE

2.5 HX711

HX711 adalah modul timbangan, yang memiliki prinsip kerja mengonversi perubahan yang terukur dalam perubahan resistansi dan mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada. Hx711 presisi 24-bit *analog-to-digital converter* (ADC) yang didesain untuk sensor timbangan digital (*weight scales*). Dikarenakan perubahan resistansi pada *straingauge* yakni dalam rentang μV , sehingga dipergunakan HX711 yang mempunyai ADC 24 bit.



Gambar 2.6 Skema HX711
(Datasheet HX711)

Gambar 2.7 Grafik Proses Konversi HX711
(Datasheet HX711)

Load cell mengirimkan hasil timbang yang berbentuk sinyal analog maka di rubah menjadi bentuk sinyal digital. DOUT dan PD_SCK mendapat masukan dari *loadcell* dimana *weight sensor module* akan mengubah dari sinyal analog

menjadi sinyal analog dengan bentuk seperti getaran pulsa. Dimana pengambilan data dari HX711 dengan komunikasi 2 data yakni data dan *clock*. Saat data atau DOUT dalam keadaan *high* maka tidak terjadi pengambilan data, saat DOUT *low* maka terjadi pengambilan data ke mikrokontroler sebagai data digital berat yang telah dikonversi.

Dalam keadaan *clock* pulsa positif ke-25-27 pada pin SCK, data digeser keluar dari DOUT sebagai pin *output*. Setiap pulsa SCK bergeser keluar sedikit, dimulai dengan bit MSB pertama sampai semua 24 bit bergeser keluar. Pulsa 25 pada *input* SCK akan menarik pin DOUT kembali dalam keadaan *high*. *Clock* pulsa SCK tidak boleh kurang dari 25 atau lebih dari 27 dalam satu periode konversi, untuk menghindari menyebabkan kesalahan komunikasi serial.

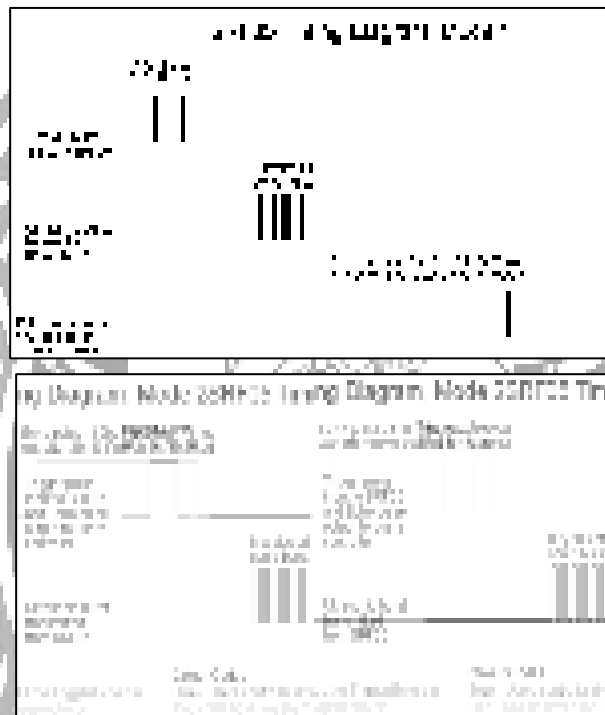
2.6 Sensor SRF-05

Sensor SRF05 merupakan sensor pengukur jarak yang menggunakan gelombang ultrasonik. Dimana prinsip kerja sensor ini adalah Pemancar (*transmitter*) mengirimkan gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek, sehingga didapat jarak sensor dengan obyek. Berikut adalah persamaan jarak pada sensor SRF-05 :

$$\text{Jarak} = \frac{\text{Kecepatan suara} \times \text{waktu pantul}}{2} \dots\dots\dots(2.2)$$

Sensor SRF-05 ini bekerja pada tegangan DC 5 volt, memiliki beban arus sebesar 30 mA – 50 mA, menghasilkan gelombang dengan frekuensi 40 KHz, memiliki jangkauan jarak dengan minimal jarak 3cm dan maksimal sebesar

400cm dan membutuhkan *trigger input* minimal sebesar 10 uS. Dalam sensor SRF-05 ini, dapat menggunakan dua pilihan mode yaitu *input trigger* dan *output echo* terpasang pada pin yang berbeda atau *input trigger* dan *output echo* terpasang dalam satu pin yang sama. Berikut adalah *timing diagram* SRF-05 pada mode 1 dan 2 :



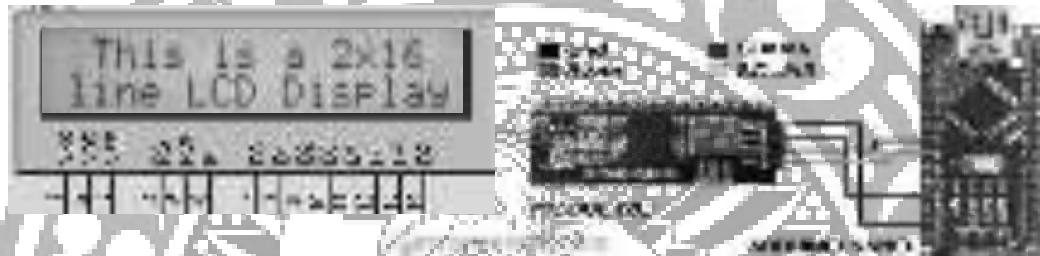
Gambar 2.8 Diagram Timing SRF-05 Mode 1 dan Mode 2

(Sumber: <https://www.robot-electronics.co.uk/htm/srf05tech.htm>)

2.7 LCD 16x2

LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada dalam bentuk karakter, huruf, angka ataupun grafik. Agar LCD dapat terhubung dengan arduino, alat ini menggunakan modul I2C sebagai komunikasi serial. Sehingga

hasil yang diperintahkan oleh mikrokontroler dapat langsung ditampilkan pada LCD. Selain itu, modul I2C ini digunakan untuk dapat menghemat *port* yang terhubung pada mikrokontroler karena jika menggunakan *port* yang ada pada LCD langsung terhubung pada mikrokontroler akan membutuhkan banyak *port* mikrokontroler. Dengan menggunakan modul ini, hanya membutuhkan 2 *port* mikrokontroler agar terhubung dengan modul. Konfigurasi pin LCD 16x2 dapat dilihat pada Gambar 2.9 berikut:



Gambar 2.9 Konfigurasi pin LCD 16x2 dan Modul I2C
(www.alldatasheet.com)

2.8 Mekanisme Kerja Alat

Alat yang kami rancang dapat digunakan untuk mengukur berat aktual yang dapat menggunakan sensor *strain gauge* dan volume barang yang dapat menggunakan sensor SRF-05. Keluaran dari sensor-sensor tersebut akan dibaca oleh Arduino UNO yang digunakan sebagai masukan agar proses untuk menentukan nilai yang terbesar dari berat aktual dan berat volume dapat diketahui. Setelah nilai yang terbesar dari kedua berat tersebut didapatkan maka hasilnya akan ditampilkan pada LCD dan dikirimkan pada labview sebagai data serial. Data serial yang diterima oleh labview berupa berat pada barang. Berat tersebut diterima dan dikalikan pada harga tiap kota yang akan ditujukan untuk proses penentuan harga.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Pembuatan alat ini akan dilakukan di laboratorium Robotika Medis Departemen Fisika, Fakultas Sains dan Teknologi Universitas Airlangga, selama kurang lebih 4 bulan, dimulai pada bulan April sampai Juli 2016.

3.2 Alat dan Bahan

Alat dan Bahan pada Tugas akhir yang digunakan agar penelitian dapat berjalan secara landar adalah sebagai berikut:

3.2.1 Alat

Alat yang dibutuhkan untuk membuat sistem ini yaitu menggunakan multimeter untuk menguji komponen dan jalur PCB, catu daya untuk memberikan supply tegangan pada alat, *downloader* IDE Arduino digunakan untuk mengunggah program ke minimum sistem Arduino UNO dan solder untuk menyolder komponen ke PCB.

3.2.2 Bahan

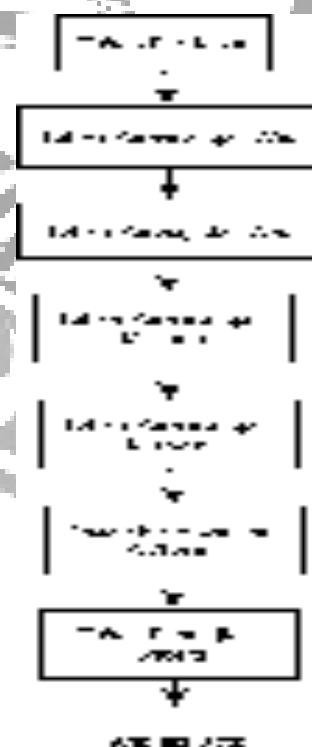
Bahan yang dibutuhkan untuk membuat alat ini adalah minimum sistem Arduino UNO digunakan sebagai kontroler pada sistem, Arduino UNO sebagai program untuk ke mikrokontroler, labview digunakan sebagai proses interface pada laptop atau PC untuk mengisikan data user dan penampilan harga, sensor *strain gauge* untuk mengukur berat pada barang, sensor SRF-05 digunakan

untuk mengukur panjang, tinggi, lebar pada barang, LCD digunakan untuk keluaran tampilan berat dan volume pada barang, *box* barang untuk simulasi barang yang akan digunakan adalah kardus. Sedangkan kayu, besi, *acrylic* dan alumunium akan digunakan untuk kerangka pada alat.

3.3 Prosedur Penelitian

Prosedur yang digunakan dalam perancangan dan pembuatan sistem terdiri dari beberapa tahapan yaitu tahap perancangan alat, tahap perwujudan alat, tahap pengujian sistem, dan analisis data.

Dari masing-masing tahapan yang akan dilakukan harus saling berkesinambungan antar tahapannya, oleh sebab itu tahapan yang dilakukan harus sesuai dengan yang diharapkan sebelum dilanjutkan ke tahap berikutnya. Berikut grafik tahapan prosedur penelitian :



Gambar 3.1 Diagram Prosedure Kerja

3.3.1 Tahap Persiapan

Tahap persiapan merupakan langkah awal penulis dalam melakukan sebuah penelitian, penulis melakukan studi literatur dengan mencari berbagai acuan pada buku, jurnal, artikel maupun tugas akhir dengan narasumber yang jelas dan terpercaya sehingga melengkapi literatur penelitian ini. Penulis juga menyiapkan alat dan bahan yang diperlukan untuk mempersiapkan dalam membuat penelitian.

3.3.2 Tahap Perancangan Alat

Tahap perancangan alat meliputi perancangan mekanik dan perancangan *hardware*. Perancangan mekanik terdiri dari sebuah kotak dengan 3 sensor SRF-05 dan 1 sensor *strain gauge*. Cara kerja dari sensor-sensor tersebut yaitu mengukur panjang, lebar, dan tinggi untuk sensor SRF-05, sedangkan sensor *strain gauge* mengukur massa benda yang tegangannya dikonversikan oleh modul HX711 menjadi keluaran digital berupa bentuk kilogram. Sensor-sensor tersebut akan dibaca serta diproses oleh arduino dan ditampilkan pada LCD berapakah besar dimensi dan berat pada barang tersebut. Labview pada pc/laptop menerima data serial untuk proses penentuan harga.

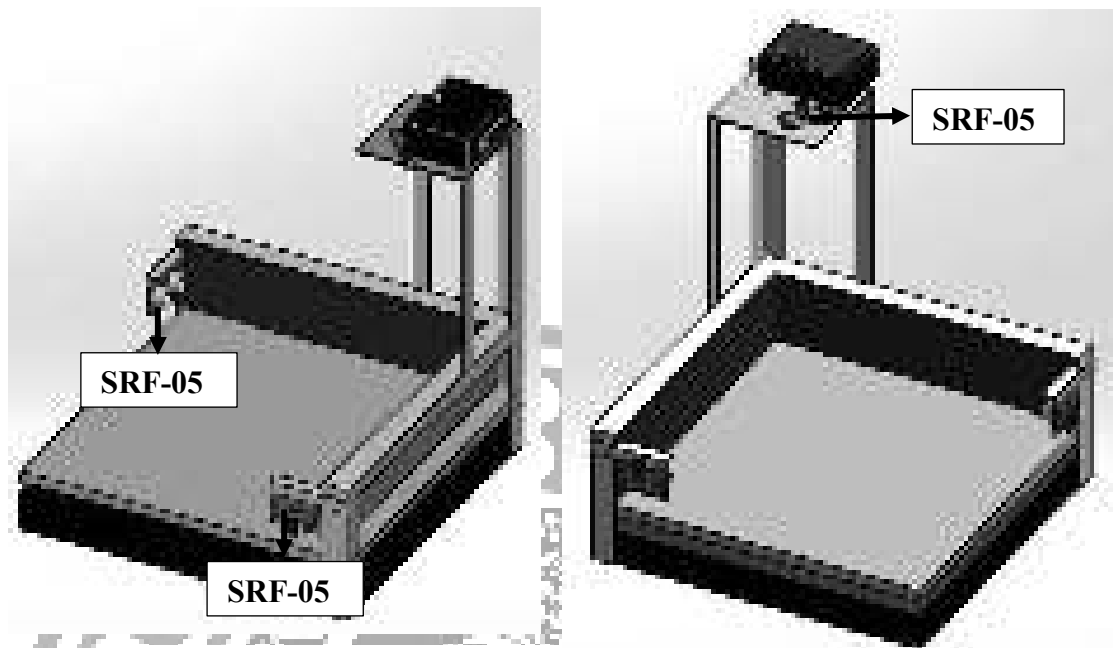
3.3.3 Tahap Perwujudan Alat

Tahap perwujudan alat ini meliputi prealisasi dari perancangan alat dan dibagi menjadi dua tahap yakni tahap perwujudan meliputi tahap perancangan mekanik dan *hardware* dari pembuatan alat yang telah dibuat, dilanjutkan dengan perancangan dan perakitan komponen –

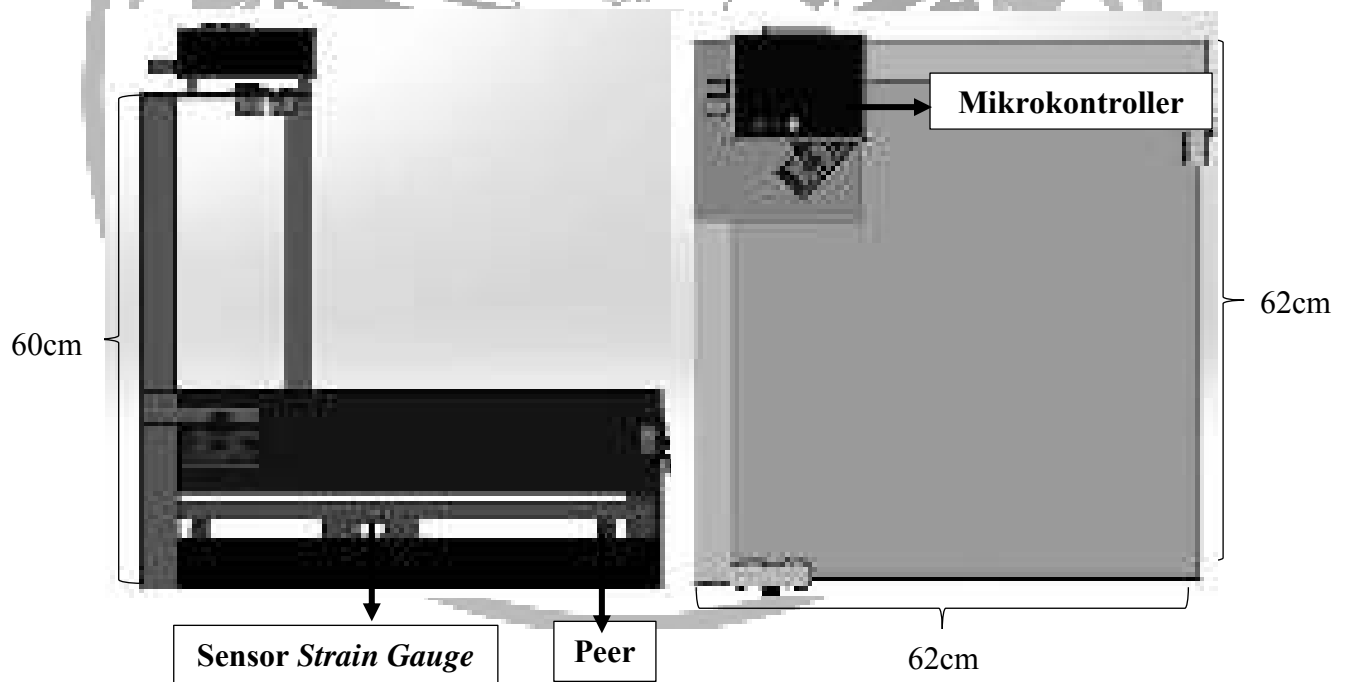
komponen elektronika yang akan membentuk suatu kesatuan sistem alat, meliputi pembuatan rangkaian minimum Arduino UNO, sensor SRF-05, sensor *strangauge*, rangkaian LCD. Dalam hal ini pemilihan komponen dapat mempengaruhi kinerja dari alat dan juga kualitas sistem yang akan dibuat. Sedangkan tahap pembuatan *software* meliputi tahap pembuatan program untuk menjalankan sistem dari alat yang dibuat. Berikut penjabaran dari masing-masing tahapan:

1. Tahap Perancangan Mekanik

Tahap pembuatan mekanik terdiri atas pembuatan rangka ruang untuk penentuan jenis berat pada barang yang terbuat dari *acrylic* dan kayu. Terdapat besi untuk penyangga dari kerangka. Sensor diletakkan pada masing-masing bagian kerangka. Mekanisme pembaca volume menggunakan 3 buah sensor SRF-05 yang diletakkan pada bagian depan untuk membaca panjang *box*, samping kiri untuk membaca lebar *box*, dan pada bagian atas untuk membaca tinggi *box*. Mekanisme penimbang untuk dapat mengetahui berat barang yang berada pada bawah tumpuan kerangka *acrylic* untuk lengan penimbang dan tempat *strain gauge*. Desain mekanik alat penentuan jenis berat ini dapat dilihat pada Gambar berikut ini



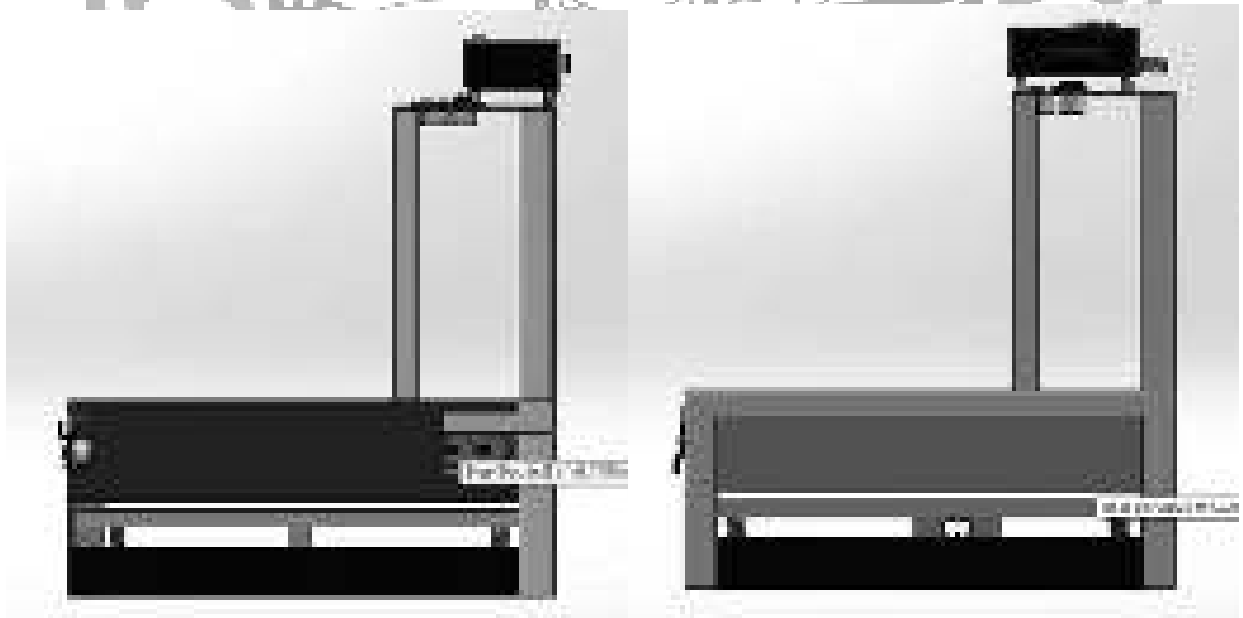
Gambar 3.2 Desain Mekanik Alat



Gambar 3.3 Desain Mekanik Tampak Bawah dan Atas



Gambar 3.4 Desain Mekanik Tampak Belakang

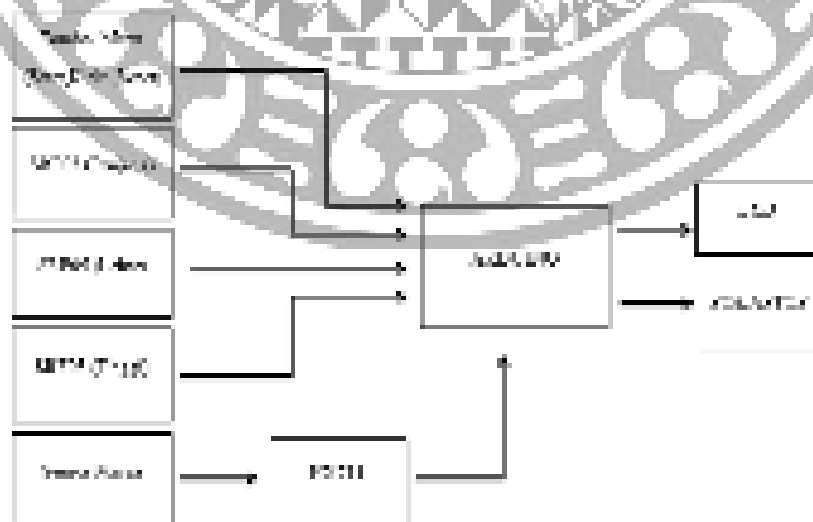


Gambar 3.5 Desain Mekanik Tampak Kanan dan Kiri

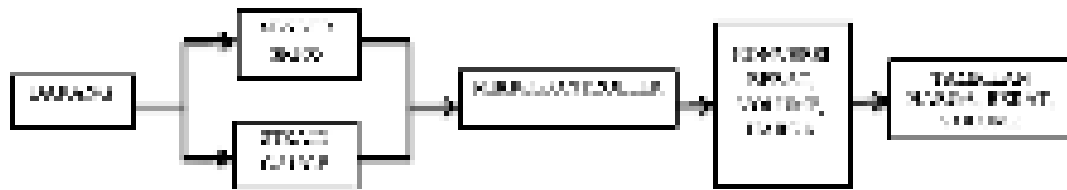
2. Tahap Perancangan Hardware

Tahap pembuatan *hardware* terdiri atas pembuatan beberapa rangkaian elektronik yang dapat menjalankan sistem penentuan jenis berat secara otomatis. Adapun rancangan *hardware* dari sistem yang akan dibuat adalah meliputi Arduino UNO, sensor SRF-05, LCD, dan *strain gauge*.

Cara kerja dari alat ini adalah dimulai dari penempatan *box* pada tempat yang telah ditentukan, yang mana selanjutnya pembacaan volume (panjang, lebar, tinggi) pada *box* didapatkan dengan adanya 3 buah sensor SRF-05. Selanjutnya Arduino UNO membaca massa pada *box* dengan menggunakan sensor massa *strain gauge*, tegangan keluaran dari *strain gauge* akan dikonversi oleh modul HX711 menjadi keluaran digital yang akan dibaca oleh Arduino UNO. LCD berguna sebagai tampilan hasil keluaran sensor *straingauge* dan SRF-05. Apabila nilai pembacaan sensor *straingauge* dan SRF-05 telah didapat, Arduino UNO akan mengirimkan data serial pada PC untuk proses penentuan harga. Diagram blok sistem otomasi penentuan harga ini dapat dilihat pada Gambar 3.6 berikut:



Gambar 3.6 Blok Diagram Alat



Gambar 3.7 Flow Control Sistem

Input : Barang

Aktuator : Sensor SRF05 Dan Sensor Strain Gauge

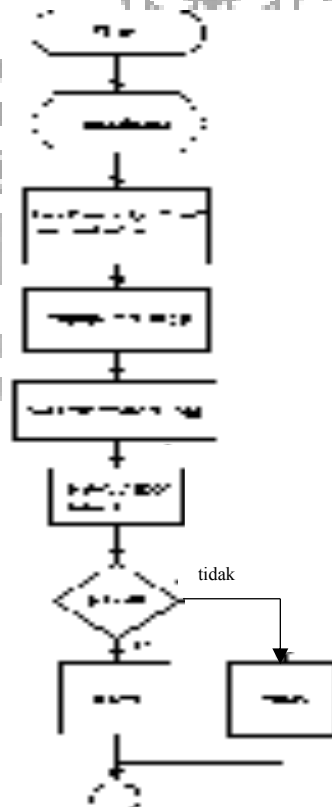
Controller : Mikrokontroler atau Arduino

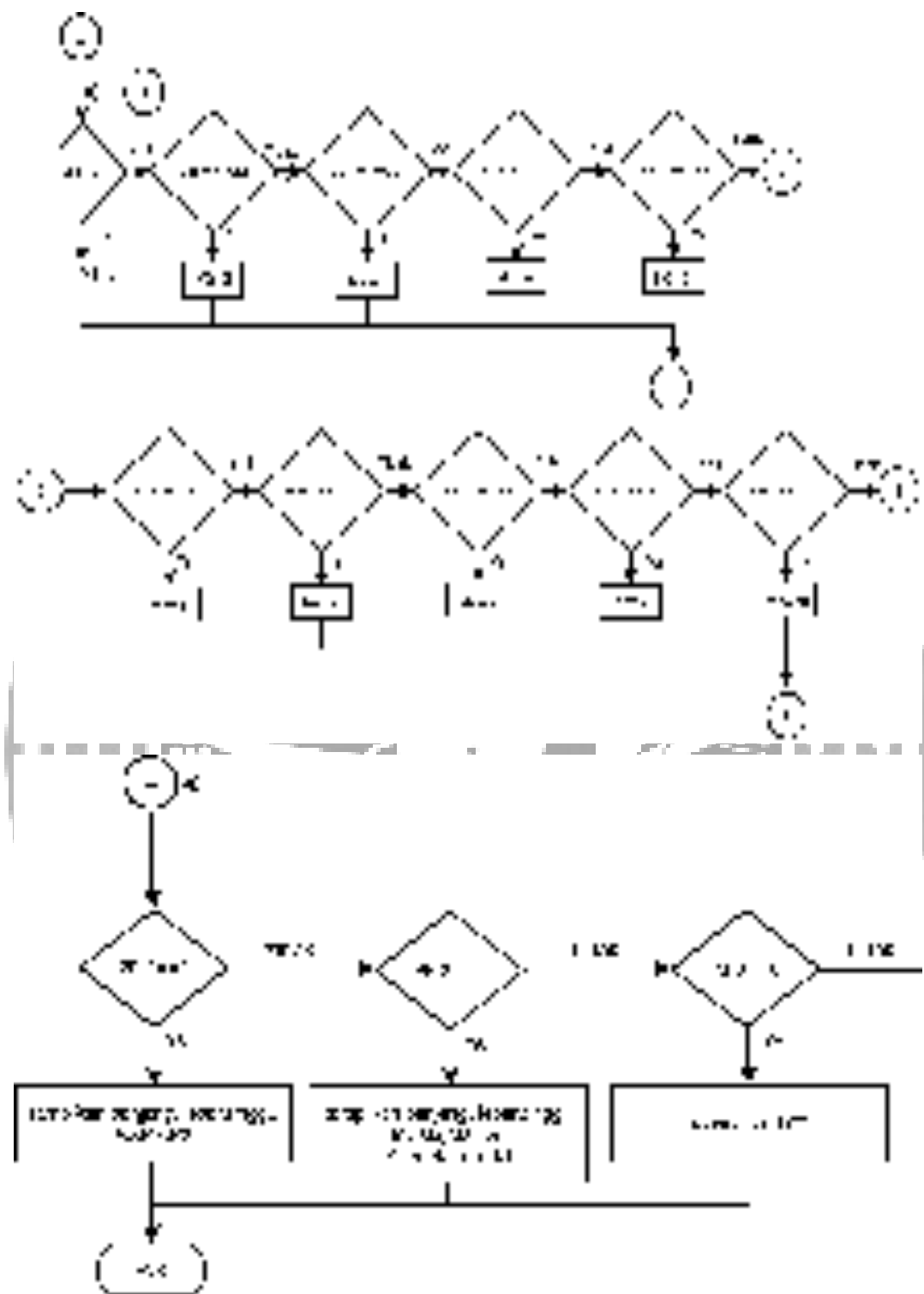
Proses : Berupa sebuah konversi untuk mendapatkan data berat, volume, harga

Output : Tampilan pada LCD dan Labview ,menampilkan harga dan volume

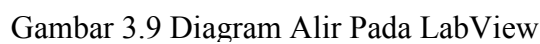
3. Tahap Perancangan Software

Tahap pembuatan *software* meliputi pembuatan program untuk mengeksekusi rancangan *hardware* yang telah dibuat. *Software* yang digunakan yakni IDE Arduino dan labview. Diagram alir program pada Arduino dan LabView ini dapat dilihat pada gambar 3.7 dan gambar 3.8 berikut :





Gambar 3.8 Diagram Alir Arduino UNO



TUGAS AKHIR

yang biasa disebut berat volume. Setelah berat volume dan berat asli telah didapat, kedua berat tersebut akan dibandingkan nilai mana yang lebih besar dan dikirimkan pada Labview jika *push button* 2 dipilih agar dapat menentukan dilakukan proses penentuan harga.

Data serial pada arduino akan dikirimkan pada Labview, kemudian Labview menerima data *ascii* dan diolah menjadi data *string* agar dapat dibaca oleh Labview. Data *string* tersebut dikonversi menjadi *number* agar dapat diproses untuk dapat dikalikan harga tiap kota dan ditampilkan pada Labview.

3.3.4 Tahap Pengujian Alat

Tahap pengujian alat terdiri dari pengujian seluruh sistem alat yang sudah dibuat yakni meliputi uji sensor SRF-05 dan uji sensor *strain gauge*. Pengujian ini dimaksudkan untuk mengetahui karakteristik dari sensor maupun sistem yang digunakan dalam penelitian ini. Berikut penjelasan masing-masing pengujian yang dilakukan.

1. Pengujian Sensor SRF-05

Dalam pengujian pada sensor SRF 05 ini, dilakukan akuisisi pengukuran jarak yang terbaca pada sensor dengan jarak sebenarnya menggunakan mistar. Sedangkan yang digunakan sebagai indikator jarak adalah *box* barang sebagai obyek. Setelah diperoleh data pengukuran, maka dapat diketahui seberapa besar nilai linieritas yang terjadi pada keluaran waktu sensor SRF -05 dengan jarak yang dibaca sensor dan simpangan yang terjadi antara menggunakan mistar dengan menggunakan

sensor SRF-05 yang dipakai dalam sistem. Semakin nilai linieritas yang dihasilkan mendekati 1 maka semakin akurat sensor yang digunakan.

a. Pengujian Linieritas sensor SRF-05 A, B dan C

Tabel 3.1 Pengujian Jarak Mistar terhadap Waktu Tempuh Sensor SRF-05 A

No.	Mistar (cm)	Pulse SRF-05	Linieritas (R^2)
1	4	260	0,9996
2	7	423	
3	10	617	
4	12	730	
5	14	824	
6	16	953	
7	18	1068	
8	21	1237	
9	24	1434	
10	26	1518	
11	29	1688	
12	31	1801	
13	33	1919	
14	35	2057	
15	38	2174	
16	41	2341	
17	43	2475	
18	46	2624	
19	50	2874	

Tabel 3.2 Pengujian Jarak Mistar terhadap Waktu Tempuh Sensor SRF-05 B

No.	Mistar (cm)	Pulse SRF-05	Linieritas (R^2)
1	3	212	0,9995
2	5	321	
3	7	430	
4	9	536	
5	11	648	
6	13	760	

7	15	896
8	17	1009
9	20	1179
10	22	1293
11	25	1480
12	27	1597
13	30	1768
14	32	1904
15	35	2059
16	37	2169
17	40	2280
18	42	2425
19	45	2572
20	47	2720
21	50	2874

Tabel 3.3 Pengujian Jarak Mistar terhadap Waktu Tempuh Sensor SRF-05 C

No.	Mistar (cm)	Pulse SRF-05	Linieritas (R^2)
1	3	200	0,9994
2	6	355	
3	8	469	
4	10	606	
5	13	757	
6	16	939	
7	18	1058	
8	20	1170	
9	23	1389	
10	26	1530	
11	28	1642	
12	30	1767	
13	33	1917	
14	36	2115	
15	38	2204	
16	40	2359	
17	43	2458	
18	46	2626	
19	48	2763	
20	50	2875	

Tabel 3.4 Linieritas Jarak Mistar Terhadap SRF-05 A Bahan Kardus

No.	Mistar (cm)	SRF-05 1 (cm)	Simpangan	Linieritas(R^2)
1	4	3.8	0.2	0.998
2	7	5.8	1.2	
3	10	9.8	0.2	
4	12	11.8	0.2	
5	14	13.8	0.2	
6	16	15.8	0.2	
7	18	18.6	0.4	
8	21	23.1	2.1	
9	24	24.2	0.2	
10	26	25.8	0.2	
11	29	28.8	0.2	
12	31	30.8	0.2	
13	33	32.5	0.5	
14	35	34.9	0.1	
15	38	37.9	0.1	
16	41	40.8	0.2	
17	43	42.8	0.2	
18	46	45.8	0.2	
19	50	49.8	0.2	
Rata-rata Simpangan			±0.36	

Tabel 3.5 Linieritas Jarak Mistar Terhadap SRF-05 B Bahan Kardus

No.	Mistar (cm)	SRF-05 1 (cm)	Simpangan	Linieritas(R^2)
1	3	2.8	0.2	0.9998
2	4	3.8	0.2	
3	5	4.7	0.3	
4	7	6.7	0.3	
5	9	8.7	0.3	
6	13	12.7	0.3	
7	15	14.9	0.1	
8	17	16.8	0.2	
9	20	20.2	0.2	
10	22	22	0	
11	25	24.9	0.1	
12	27	26.7	0.3	
13	30	29.8	0.2	
14	32	31.8	0.2	

15	34	33.6	0.4
16	36	35.6	0.4
17	39	38.7	0.3
18	41	40.7	0.7
19	44	43.2	0.8
20	46	45.2	0.8
21	49	48.2	0.8
22	50	49.1	0.9
Rata-rata Simpangan			±0.36

Tabel 3.6 Linieritas Jarak Mistar Terhadap SRF-05 C Bahan Kardus

No.	Mistar (cm)	SRF-05 I (cm)	Simpangan	Linieritas(R^2)
1	3	3	0	0.9997
2	6	5.8	0.2	
3	8	7.9	0.	
4	10	9.8	0.	
5	13	12.7	2.	
6	16	16.6	0.	
7	18	18	0	
8	20	20	0	
9	23	22.6	0.4	
10	26	25.6	0.4	
11	28	27.5	0.5	
12	30	29.5	0.5	
13	33	32.5	0.5	
14	36	35.5	0.5	
15	38	36.9	1.1	
16	40	39.8	0.2	
17	43	42.3	0.7	
18	46	45.3	0.7	
19	48	47.3	0.7	
20	50	49.6	0.4	
Rata-rata Simpangan			±0.5	

Tabel 3.7 Linieritas Jarak Mistar Terhadap SRF-05 A Bahan Plastik

Mistar (cm)	SRF-05 A	R^2
4	3.8	0,997
7	6.66	
10	10	
12	12	
14	13.6	
16	16	
18	17.6	
21	20.6	
24	24.2	
26	26.2	
29	29.1	
31	31	
33	33.5	
35	35.5	
38	38.5	
41	42.4	
43	43.3	
46	46.5	
50	48	

Tabel 3.8 Linieritas Jarak Mistar Terhadap SRF-05 B Bahan Plastik

Mistar (cm)	SRF-05 B	R^2
3	3.8	0,9994
4	2.7	
5	4.8	
7	6.8	
9	8.8	
13	12.8	
15	14.7	
17	16.8	
20	19.7	
22	21.7	
25	24.7	
27	26.8	
30	29.8	
32	31.8	
34	34.6	
36	35.8	
39	39.1	
41	41	
44	44	
46	46	
49	48.9	
50	50.4	

Tabel 3.9 Linieritas Jarak Mistar Terhadap SRF-05 C Bahan Plastik

Mistar (cm)	SRF-05 C	R^2
3	2.9	0,9996
6	6.8	
8	7.9	
10	9.8	
13	12.8	
16	16.7	
18	18	
20	20.3	
23	23.3	
26	25.8	
28	27.9	
30	29.8	

33	32.9	
36	35.8	
38	37.7	
40	39.85	
43	42.4	
46	45.8	
48	47.8	
50	49.7	

2. Pengujian Sensor *Strain Gauge*

Pengujian linieritas pada sensor *strain gauge* dilakukan dengan cara melakukan perbandingan pembacaan massa yang terbaca pada kalibrator yang digunakan dengan massa yang terbaca pada sensor *strain gauge* yang digunakan. Kalibrator yang digunakan adalah timbangan digital dengan kapasitas maksimal 20 Kg dengan menggunakan beban timbal yang sama. Dari melakukan perbandingan tersebut maka dapat diketahui seberapa besar nilai linieritas dan simpangan yang terjadi antara menggunakan timbangan kalibrator dengan menggunakan sensor *strain gauge* yang dipakai dalam sistem. Semakin nilai linieritas yang dihasilkan mendekati 1 maka semakin akurat sensor yang digunakan.

Tabel 3.10 Pengujian Linieritas Sensor *Strain Gauge*

No.	Berat Kalibrator (kg)	Berat <i>Strain Gauge</i> (kg)	Simpangan	Linieritas (R^2)
1	0.7	0.6	0.1	0,9988
2	1	1	0	
3	1.7	1.6	0.1	
4	2.6	2.8	0.2	
5	3.3	3.5	0.2	
6	3.6	3.8	0.2	
7	4.3	4.3	0	

8	5.9	6.2	0.3
9	6.6	6.9	0.3
10	6.9	7	0.1
11	7.6	7.7	0.1
12	8.5	8.7	0.2
13	9.2	9.4	0.2
14	9.5	9.7	0.2
Rata-rata Simpangan			± 0.15

3.3.5 Analisis Data

Pengambilan data ini dilakukan untuk mengetahui seberapa efektif *software* dan *hardware* yang telah dibuat sehingga alat ini dapat bekerja sesuai dengan harapan. Untuk menguji kelayakan maupun keberhasilan sistem yang telah dibuat apakah sesuai dengan harapan atau tidak maka dapat dilihat dari data pengujian linieritas sensor dengan kalibrator dan analisis data yang akan diambil. Data pengujian ini terdiri dari beberapa data yakni hubungan keluaran tegangan pada *strain gauge* dengan massa kalibrator timbangan digital, hubungan keluaran kalibrator massa dengan keluaran sensor *straingauge* menggunakan IC HX711 dan hubungan antara jarak menggunakan mistar dengan jarak yang terbaca oleh sensor SRF-05. Data pengujian linieritas tersebut sudah didapatkan pada uji coba sebelumnya.

Sedangkan untuk data yang akan dianalisis dalam sistem ini yakni terdiri atas hubungan antara lamanya waktu dengan jumlah barang yang akan diproses pada *system*. Selain itu, uji coba keseluruhan sistem dapat

dianalisis untuk mengetahui apakah sistem yang telah dibuat dapat mengukur dan menimbang dengan baik.

3.11 Tabel Lama waktu Mengukur dan Menimbang Barang

Waktu mengukur dan Menimbang (s)	Jumlah Barang Menggunakan Sistem	Jumlah Barang Menggunakan cara Manual
60	2	2
120	5	4
180	8	5

Tabel 3.12 Pengujian Keseluruhan Sistem

Percobaan ke-	Berat Aktual Asli (kg)	Berat Aktual Terukur (kg)	Simpangan	Berat Volume Asli (kg)	Berat Volume Terukur (kg)	Simpangan
1	2.8	2.6	0.2	0.7	0.6	0.1
2	5.9	5.7	0.2	1.2	1.1	0.1
3	0.4	0.5	0.1	1.4	1.23	0.17
4	0.3	0.5	0.2	4.2	3.7	0.5
5	1	1.5	0.5	2.6	2.3	0.3
6	0.8	0.8	0	1	0.8	0.2
7	0.3	0.5	0.2	3.7	3.7	0
8	0.2	0.4	0.2	2.9	2.8	0.1
Rata-rata simpangan			±0.2	Rata-rata Simpangan		±0.18

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pembuatan Perangkat Lunak (Software)

Pada sub bab ini akan membahas tentang pembuatan dan pengujian perangkat lunak (*software*) pemograman maupun *interface* pada otomatisasi penentuan harga barang berdasarkan berat dan volume. Sesuai dengan perangkat lunak yang dibuat, perangkat lunak ini terdiri dari sub program yang terdiri dari sub awal program, program pengaturan awal, program data, program rata-rata, program penentuan harga, program reset, program tampil, program perulangan, dan sub bab tampilan interface penampilan harga pada LabView.

4.1.1 Sub Program Awal

Pada sub bab ini akan membahas mengenai program awal yang akan digunakan sebagai salah satu fungsi yang hanya satu kali eksekusi ketika awal program berjalan. Biasanya berisikan inisialisasi fungsi-fungsi yang digunakan dalam program yang diberikan pada sistem. Berikut program yang digunakan dapat dilihat dibawah ini:

```
#include<Wire.h>           //fungsi pustaka wire
#include<LiquidCrystal_I2C.h> //fungsi pustaka komunikasi I2C LCD
#include<LiquidCrystal.h>   //fungsi pustaka LCD
#include<HX711.h>           //fungsi pustaka HX711
LiquidCrystal_I2C lcd (0x27,16,2); //LCD address 0x27 untuk jenis
                                   LCD dengan tampilan 16x2

#define echo1 3              // pin 3 dan 2 sebagai port echo1
                             }
#define trig1 2              dan trigger SRF 1
```

```

...

#define DOUT A0                // Port DOUT pin A0 Sensor
SG

#define SCK A1                // Port SCK pin A1 Sensor
SG

#define pb1 8                  //pin 8 sebagai push button 1
...

HX711 scale (A0,A1);          //inisialisasi perhitungan
                                scale port (A0,A1)
int calibration_factor = -7140; //kalibrasi nilai scale untuk
                                0kg
float cm1,cm2,cm3,v,volume;   //variabel float data cm1, cm2,dll
float A,M,M2                   //variable float perhitungan berat,
                                nilai berat, dan penentuan berat
int pulse1,pulse2,pulse3;     //variabel integer waktu tempuh
int rerata1,rerata2,rerata3;   //variabel integer untuk rata-rata
                                jarak;
int i,b,n;                     //variable integer penambahan rerata
float panjang, lebar, tinggi  //variable float merata-rata
int x,y,z;                     //variabel integer untuk pb1-3
int step;                      //variabel awal program

void setup () {
step=0;
scale.set_scale();             //sebagai kalibrasi scale untuk
                                berat 0 kg

scale.tare();                   //Reset agar nilai scale ke 0

lcd.init();                     //inisialisasi LCD

lcd.backlight();               // inisialisasi nyala LCD

lcd.begin(16,2);               // inisialisasi jenis LCD 16x2

scale.set_scale();             //sebagai kalibrasi scale untuk
                                berat 0 kg

```

```

scale.tare();                      //Reset agar nilai scale ke 0
pinMode(trig1, OUTPUT);           //inisialisasi mode pin trig1 sebagai
                                  output
pinMode(echo1, INPUT);            //inisialisasi mode pin echo1 sebagai
                                  input
....

```

Penjelasan :

Pada sub program ini, pertama-tama melakukan inisialisasi pustaka jalannya program. Pada program LCD, dapat menggunakan menggunakan jalur *wire* dan jalur komunikasi I2C. Saluran atau bus pada komunikasi serial ini dikenal dengan nama SCL dan SDA. SCL atau Serial *clock* berfungsi untuk menyelaraskan data yang ada antara *master* dan *slave*. Sedangkan SDA atau Serial Data berfungsi sebagai saluran data. Pada program HX711 untuk membaca berat pada sistem dilakukan inisialisasi pustaka HX711. Selain inisialisasi diatas, masing-masing *port* yang digunakan pada program harus diinisialisasi dan memberikan variabel sesuai dengan fungsinya.

4.1.2 Sub Pengaturan Awal Kerja Program

Pengaturan awal kerja program digunakan untuk mengatur keluaran dan membaca masukkan dari sensor sesuai dengan perintah.

Berikut adalah program yang digunakan:.

```

void atur(){
  if (step==0){                  //dalam keadaan awal
    scale.set_scale(calibration_factor);} // untuk mengatur kalibrasi
                                          scale 6200 agar nilai awal
                                          0kg
    digitalWrite (trig1,LOW);          //matikan pin trig1
    digitalWrite (trig2,LOW);          //matikan pin trig2

```

```

digitalWrite (trig3,LOW);           //matikan pin trig3
delayMicroseconds (2);             //jeda selama 2 microseconds
digitalWrite (trig1,HIGH);          //menyalakan trig1
digitalWrite (trig2,HIGH);          //menyalakan trig2
digitalWrite (trig3,HIGH);          //menyalakan trig3
delayMicroseconds (10);            //jeda selama 10 microseconds

```

Penjelasan :

Pada sub program diatas, void atur digunakan untuk memerintahkan kondisi menyala pada setiap port trigger sensor SRF-05 untuk mengirimkan 8 siklus gelombang selama 10 μ s secara terus menerus hingga pin echo pada SRF-05 menerima gelombang tersebut untuk dieksekusi pada mikrokontroler.

4.1.3 Sub Program Data

Pada sub bab ini akan membahas mengenai pembacaan data berupa pembacaan massa menggunakan sensor *strain gauge* dan pengukuran jarak menggunakan sensor SRF. Berikut program yang digunakan :

```

void data(){
  digitalWrite (trig1,LOW);           //proses perhitungan
  pulse1=pulseIn(echo1,HIGH);         panjang
  cm1 = (62-(0.0177 * pulse1) - 0.8209);
  digitalWrite (trig2,LOW);           //proses perhitungan
  pulse2=pulseIn(echo2,HIGH);         lebar
  cm2 = (60-(0.0175*pulse2)) - 0.6491;
  digitalWrite (trig3,LOW);           //proses perhitungan
  pulse3=pulseIn(echo3,HIGH);         tinggi
  cm3= (60-(0.0175 * pulse3) - 0.5141);
}

```



```
A=(scale.get_units()/10);           //proses pencarian berat
```

Penjelasan :

Berdasarkan listing program di atas, setelah awal kerja program pada penjelasan sebelumnya ketika obyek telah terdeteksi, gelombang diterima pin *echo* dan akan bernilai *high* yang berarti mikrokontroler menghitung lama waktu (*pulseIn*) gelombang yang diterima oleh pin *echo*. Panjang, lebar dan tinggi benda didapatkan dari besar *pulse* lalu dikonversi dengan menggunakan rumus cm1, cm2, cm3.

Program awal untuk dapat menghitung berat dengan menginisialisasi port SCK dan DOUT sebagai jalur komunikasi data dengan mikrokontroler dan pustaka HX711 untuk menghitung berat. Kalibrasi nilai skala dalam program digunakan untuk mencari nilai kalibrasi agar nilai berat benda sesuai dengan kalibrator. Skala yang digunakan yaitu sebesar 7140. *Scale tare* digunakan agar berat kembali ke 0 ketika tidak ada benda. Untuk mencari berat benda dapat menggunakan *scale.get_units* untuk mengetahui berapa berat benda yang diletakkan pada sistem.

4.1.4 Sub Program Rerata

Pada sub program ini digunakan untuk merata-rata hasil dari pengonversian jarak menjadi lebih akurat. Percobaan dilakukan penjumlahan sebanyak 9 kali data hasil jarak lalu dirata-rata. Berikut adalah program pada sub ini:

```
void rata(){
```

```

int i;
for (i=0;i<10;i++){
    rerata+=cm1;
}
panjang=rerata1/10;
rerata1=0;
....

```

//perhitungan rata-rata 9x
hasil setiap jarak SRF1

4.1.5 Sub Program Penentuan Berat

Pada sub bab ini akan membahas mengenai eksekusi penentuan berat dengan membandingkan berat asli dengan massa volume pada barang. Berikut program yang digunakan dapat dilihat dibawah ini:

```

void penentuanberat () {
    v= (panjang*lebar*tinggi)
    ;volume= v/6000;
    //perhitungan berat volume
    if (A>volume){M=A;}
    else (A<volume){M=volume;}
    // Penentuan Berat pada Barang
    if (M<=1.2){M2=1;}
    else if (M>=1.3 && M<=2.2){M2=2;}
    else if (M>=2.3 && M<=3.2){M2=3;}
    else if (M>=3.3 && M<=4.2){M2=4;}
    ....
}

```

//Toleransi
pembulatan berat
barang untuk proses
penentuan harga

Penjelasan :

Massa volume didapatkan pada program perhitungan volume dibagi 6000 dan hasilnya dapat dibandingkan langsung dengan berat aktual. Nilai yang lebih besar tersebut akan digunakan untuk proses penentuan harga. Nilai berat volume dan aktual memiliki variabel *float* karena jika diberikan variabel integer maka memiliki

selisih nilai yang besar antara sensor dengan kalibrator. Namun jika nilai float tersebut tidak dapat dibiarkan, karena pada jasa pengiriman jika berat benda memiliki lebih dari 0,2 maka dibulatkan ke nilai berikutnya.

4.1.6 Sub Program Reset dan Tampil Pada LCD

Pada sub program ini akan digunakan untuk mengkosongkan kembali ulang dan menampilkan kembali hasil data dan penentuan berat pada LCD. Berikut program yang akan digunakan seperti yang dibawah ini:

```
void rst(){
    lcd.setCursor(0,0);
    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(" ");
}
//program untuk mereset
//agar LCD kosong

void tampil(){
    //digunakan untuk menampilkan
    //hasil pada kursor LCD

    lcd.clear();
    lcd.setCursor (0,0);
    lcd.print(panjang);
    lcd.setCursor (6,0);
    lcd.print(lebar);
    .....
}
```

Menampilkan hasil nilai pada lcd.print sesuai dengan kursor yang telah diatur

4.1.7 Sub Program Perulangan

Pada sub program ini digunakan ntuk mengeksekusi program data, program rata-rata, program penentuan harga, program *reset* dan program tampil agar dapat dieksekusi untuk berjalan secara berurutan. Berikut

program yang akan digunakan seperti yang dibawah ini:

```
void loop () {
  data();
  rata();
  penentuanberat();
  atur();
  if(z==LOW){rst();}
  if (x==LOW){tampil();}
  if (y==LOW)
  {
    lcd.clear();
    lcd.setCursor (0,0);
    lcd.print(panjang);
    lcd.setCursor (7,0);
    lcd.print(lebar);
    ...
    Serial.print(panjang);
    Serial.print(" ");
    Serial.print(lebar);
    Serial.print(" ");
    ...
  }
}
```

Menampilkan hasil nilai pada
lcd.print sesuai dengan kursor
yang telah diatur

Program untuk mengirimkan
data serial pada labView

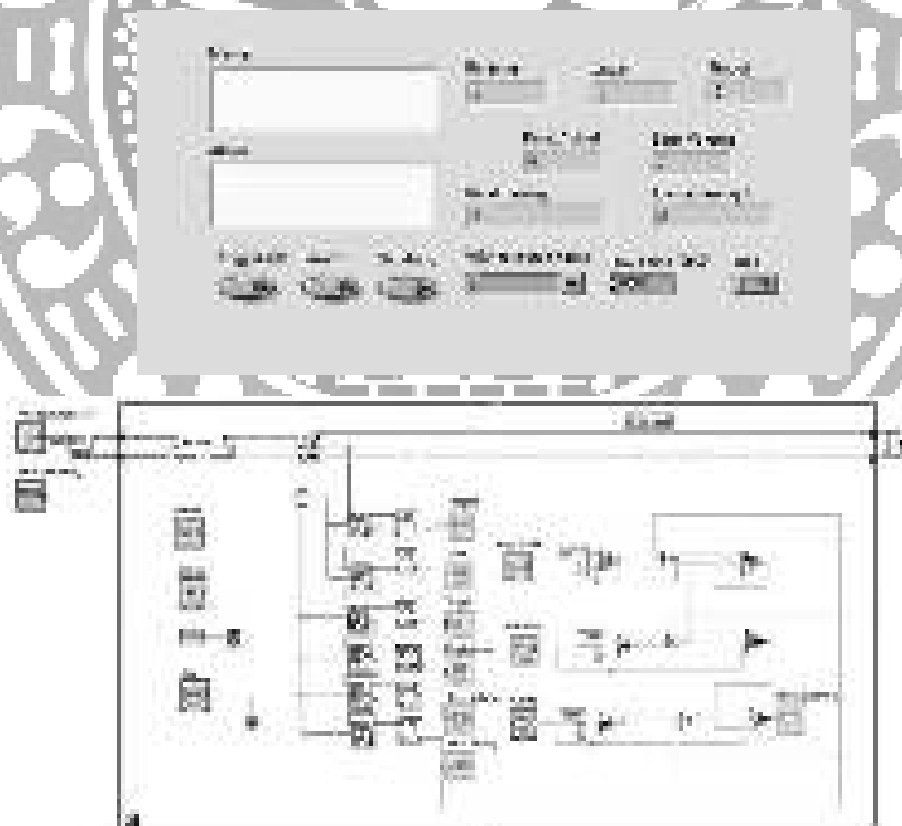
Penjelasan :

Pada sub program perulangan ini digunakan untuk mengeksekusi jalannya program mulai dari program atur pada kerja awal program, pembacaan data, dilanjutkan pada program rata dan penentuan berat sesuai dengan fungsinya. Pada program tampil dan atur, akan digunakan untuk *push button* untuk dijalankan sesuai perintah *push*

button tersebut. Jika *push button* 1 ditekan maka akan menampilkan hasil data dan penentuan berat pada LCD. Jika *push button* 2 ditekan maka akan digunakan untuk menampilkan pada LCD dan mengirimkannya pada LabView. Selanjutnya, jika *push button* 3 ditekan, maka lcd akan mereset atau mengkosongkan kembali LCD tersebut.

4.1.8 Interface Sistem Menggunakan Labview

Labview digunakan untuk menampilkan hasil data keseluruhan yang telah dicoba menggunakan mikrokontroller. Hasil akhir dari proses penentuan harga ini menampilkan hasil harga pada setiap kota tujuan. Berikut adalah tampilan *interface* pada labview :



Gambar 4.1 Gambar Tampilan *Interface* Sistem Penentuan Harga

4.2 Pembahasan Pengujian Sensor SRF-05

Sensor yang digunakan dalam pembuatan alat ini dalam mendapatkan nilai panjang, lebar, tinggi adalah sensor SRF-05 yang merupakan sensor yang mengirimkan seberkas gelombang ultrasonik lalu diukur lamanya waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Jarak yang didapat dapat menggunakan rumus

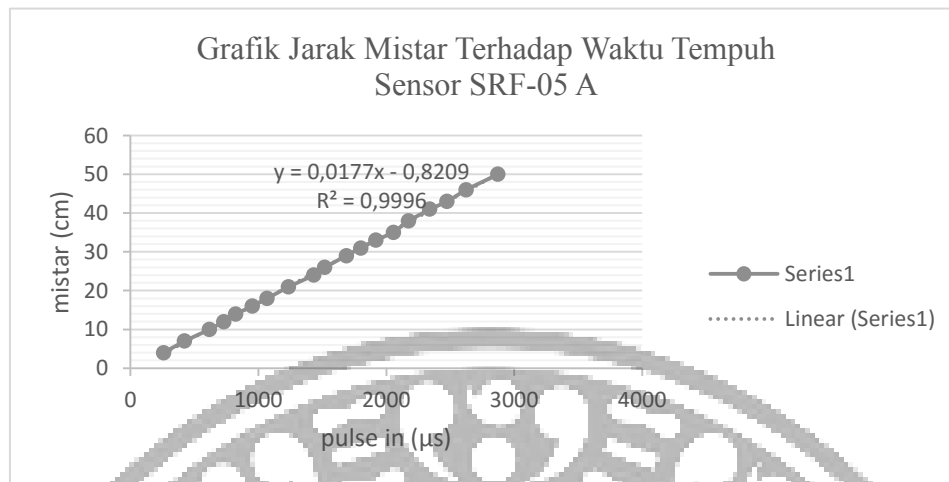
$$s = \frac{(v * t)}{2}$$

dimana v merupakan kecepatan udara dan t merupakan lamanya waktu yang diterima sensor berupa *pulse* dari pantulan benda.

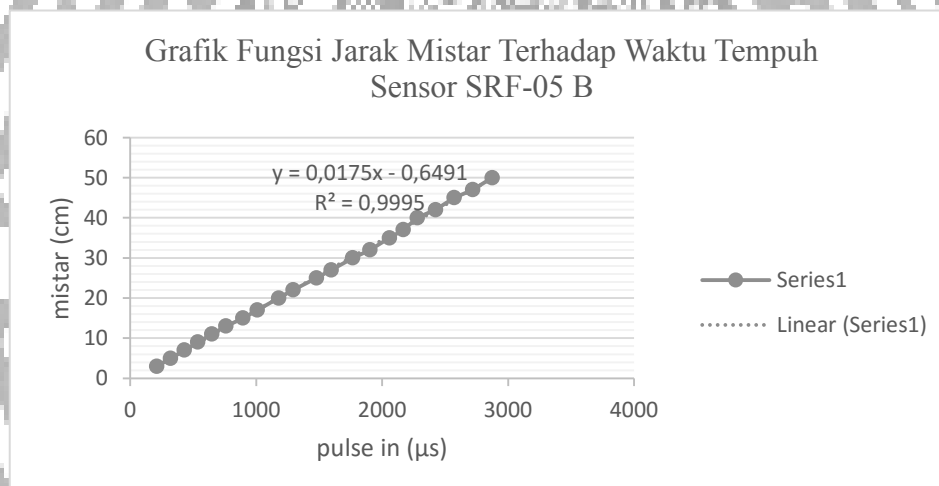
Pengujian yang ditunjukkan pada tabel 3.1, 3.2 dan 3.3 sensor SRF-05 dilakukan dengan cara melakukan perbandingan pembacaan jarak dengan menggunakan kalibrator lainnya berupa mistar dengan panjang hingga 50cm yang ditempatkan secara sejajar. Jarak yang terbaca oleh mistar digunakan sebagai data kalibrasi sensor SRF-05 untuk mendapatkan hubungan karakteristik antara jarak pada mistar dan nilai *pulse* yang terbaca oleh sensor dalam keadaan sistem melakukan pembacaan nilai jarak. Pengambilan data ini dilakukan secara bervariasi yang terbaca oleh mistar.

a. Pengujian Hubungan Jarak Mistar terhadap Waktu Tempuh Sensor

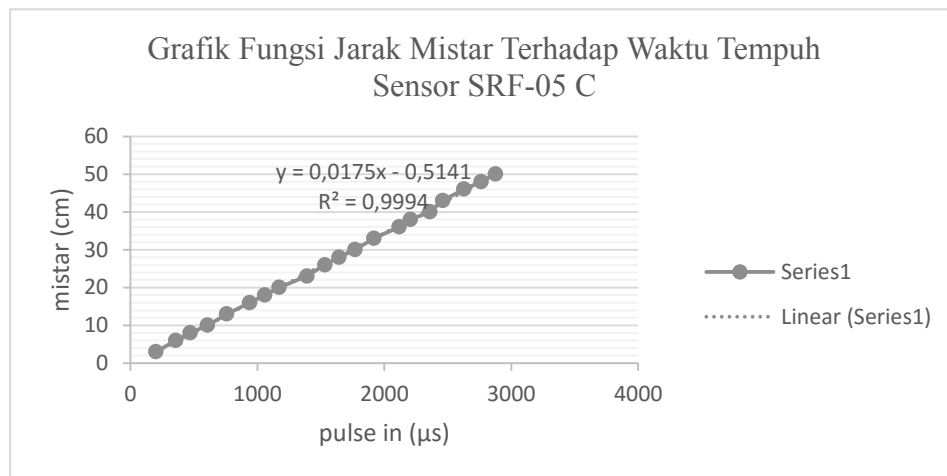
Hasil data yang diperoleh dari tiga tabel 3.1, 3.2 dan 3.3, maka dibuatlah linieritas hubungan antara mistar dengan *pulse* keluaran sensor SRF-05 A, B dan C yang dapat dilihat pada gambar grafik :



Gambar 4.2 Grafik Jarak Mistar Terhadap Waktu Tempuh (*pulse in*) sensor SRF-05 A



Gambar 4.3 Grafik Jarak Mistar Terhadap Waktu Tempuh (*pulse in*) sensor SRF-05 B



Gambar 4.4 Grafik Jarak Mistar Terhadap Waktu Tempuh (*pulse in*) sensor SRF-05 C

Pada gambar grafik 4.2, 4.3 dan 4.4, dapat diketahui bahwa pengambilan data tiap jarak akan dibandingkan dengan nilai *pulse* dari keluaran sensor SRF-05 A, B dan C untuk panjang, lebar dan tinggi sebagai data kalibrasi, sehingga menghasilkan grafik hubungan antara pembacaan jarak dengan nilai *pulse* yang dimana hasil persamaan dari grafik digunakan sebagai konversi jarak pada sensor SRF-05 A, B, dan C

Tabel 4.1 Hasil Persamaan Pada Grafik Sensor SRF-05 A, B dan C

Sensor SRF-05	Persamaan pada Grafik	R^2	Persamaan Hasil Konversi
A	$y = 0.0177x - 0.8209$	0,9996	Jarak A = (<i>pulse</i> A x 0.0177) – 0.8209
B	$y = 0.0175x - 0.6491$	0,9995	Jarak B = (<i>pulse</i> B x 0.0175) – 0.6491
C	$y = 0.0175x - 0.5141$	0,9994	Jarak C = (<i>pulse</i> C x 0.0175) – 0.5141

Persamaan hasil konversi yang dapat dilihat pada tabel 4.1. Hasil persamaan tersebut dapat digunakan untuk mengetahui hubungan waktu

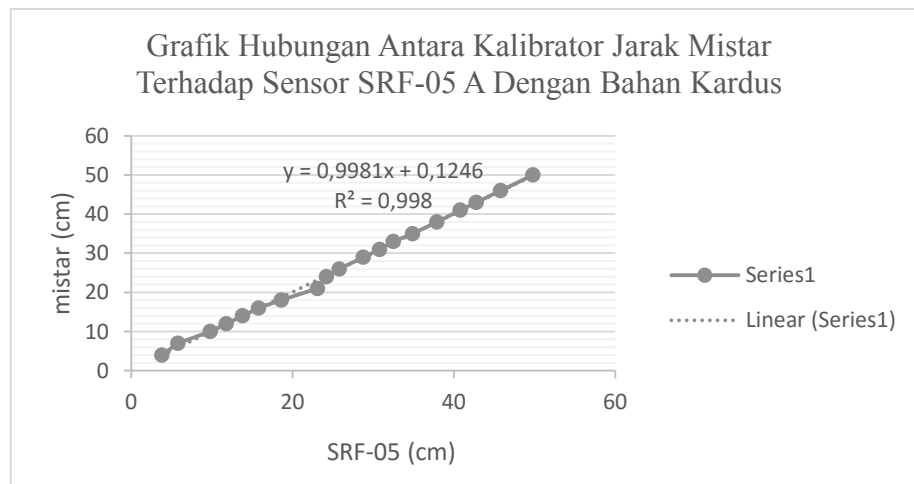
tempuh sensor A, B dan C dengan keluaran jarak pada sensor. Persamaan hasil konversi tersebut dapat langsung menjadi nilai jarak dalam satuan sentimeter pada arduino agar nilai jarak yang dihasilkan oleh sensor dapat sesuai dengan kalibrator mistar.

b. Pengujian Linieritas dan Simpangan Jarak Mistar Terhadap SRF-05

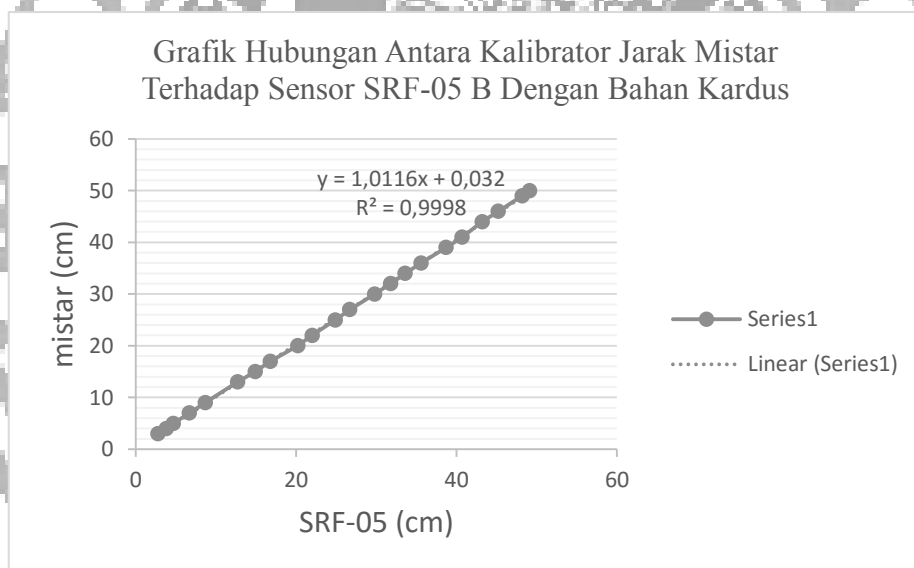
Bahan Kardus

Pengambilan data berikutnya adalah dengan mengukur jarak secara bertahap, guna untuk mengetahui akurasi dari mekanisme pengambilan jarak yang telah dibuat. Dalam pengambilan data ini jarak yang digunakan sama dengan sebelumnya yaitu 3cm hingga 50cm menggunakan bahan kardus pada benda. Hasil data dapat dilihat pada tabel 3.5, 3.6 dan 3.7 dengan data yang dihasilkan merupakan hasil linieritas antara kalibrator jarak menggunakan mistar dengan sensor SRF-05 A, B dan C yang telah dikalibrasi dengan menggunakan hasil konversi pada tabel 4.1.

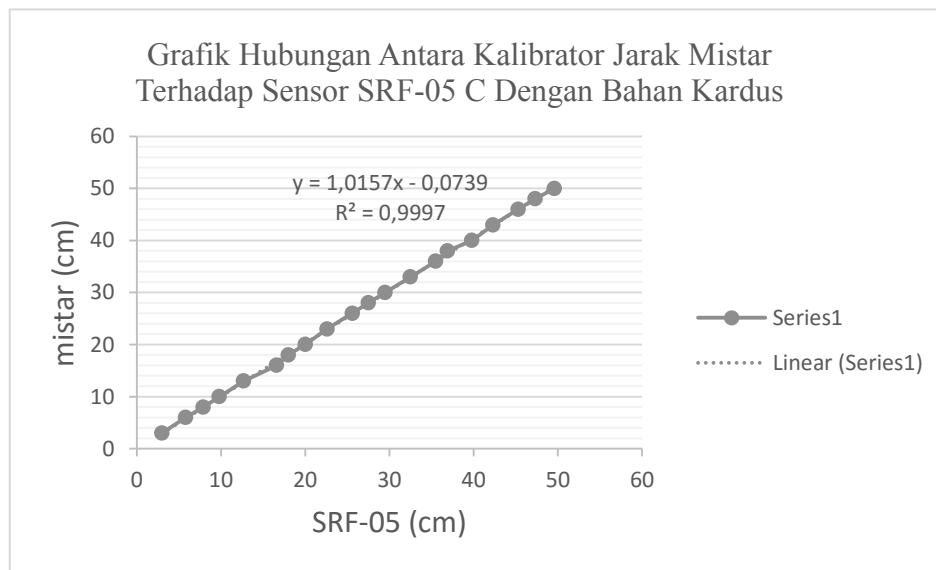
Rata-rata simpangan yang dihasilkan dari ke tiga sensor SRF-05 A, B, C pada tabel 3.5, 3.6 dan 3.7 adalah serbesar ± 0.36 untuk sensor A, ± 0.36 untuk sensor B dan ± 0.5 untuk sensor C. Karena simpangan atau *error* yang dimiliki ketiga sensor tersebut sangat kecil, maka sensor dapat digunakan dengan baik. Kemudian dibuatlah linieritas hubungan antara kalibrator jarak mistar terhadap sensor SRF-05 A, B dan C yang ditunjukkan pada gambar grafik berikut :



Gambar 4.5 Grafik Hubungan Antara Kalibrator Jarak Mistar Terhadap Sensor SRF-05 A



Gambar 4.6 Grafik Hubungan Antara Kalibrator Jarak Mistar Terhadap Sensor SRF-05 B



Gambar 4.7 Grafik Hubungan Antara Kalibrator Jarak Mistar Terhadap Sensor SRF-05 C

Berdasarkan gambar grafik 4.5, 4.6 dan 4.7 yang dihasilkan maka dapat diketahui bahwa hubungan antara jarak mistar dengan sensor SRF-05 menghasilkan nilai linieritas dengan sensor A menghasilkan $R^2 = 0,998$; sensor B menghasilkan $R^2 = 0,9998$; sensor C menghasilkan $R^2 = 0,9997$. Dengan ketiga nilai linieritas pada masing-masing sensor memiliki nilai yang mendekati 1, maka nilai linieritas yang mendekati 1 tersebut menunjukkan bahwa ke 3 sensor SRF yang digunakan sudah bekerja dengan baik dan sesuai dengan nilai hasil pembacaan kalibrator lain.

c. Pengujian Linieritas dan Simpangan Jarak Mistar Terhadap SRF-05 Bahan Kardus

Pengujian sensor SRF-05 A, B dan C terhadap bahan juga dilakukan guna untuk mengetahui bahan apakah yang dapat digunakan dan lebih baik terhadap sistem. Bahan kedua yang digunakan terhadap sistem ini menggunakan plastik dapat dilihat pada tabel 3.8, 3.9 dan 3.10.

Hasil pada tabel 3.8, 3.9 dan 3.10 maka dapat diketahui bahwa pantulan jarak sensor SRF-05 dapat bekerja lebih baik terhadap bahan kardus dengan nilai linieritas pada sensor A sebesar $R^2 = 0.998$; pada sensor B sebesar $R^2 = 0.9998$ dan pada sensor C sebesar $R^2 = 0.9997$. sedangkan dari table 4.3 diatas, bahan plastik menunjukkan nilai linieritas pada sensor A sebesar $R^2 = 0.997$; pada sensor B sebesar $R^2 = 0.9994$ dan pada sensor C sebesar $R^2 = 0.9996$ Hal ini menunjukkan bahwa bahan kardus lebih stabil untuk digunakan pada sistem daripada menggunakan bahan plastik.

4.3 Pengujian Sensor *Strain Gauge*

Sensor berat yang digunakan dalam pembuatan alat ini adalah sensor *Strain gauge* dengan beban maksimal 20kg yang memiliki *output* langsung berupa data digital, karena dalam pengujian menggunakan modul hx711 yang digunakan dalam alat ini memiliki nilai 24bit (16777216) sebagai konversi dalam bentuk keluaran kilogram.

Pengujian linieritas pada sensor *Strain gauge* dilakukan dengan cara melakukan perbandingan pembacaan berat dengan menggunakan kalibrator lainnya berupa timbangan digital. Pengambilan data ini dilakukan dengan memvariasikan berat yang terbaca oleh timbangan digital. Berikut data hasil hubungan antara pembacaan timbangan digital dengan sensor *strain gauge*:

Dari hasil pengujian Tabel 3.11 diketahui bahwa mekanisme pengujian massa dilakukan dalam kondisi sistem penimbangan, dimana rentang kondisi massa mulai dari 0.7 kg - 9.5 kg. Sehingga sensor massa

yang digunakan dalam alat ini memiliki ketelitian sebesar 2,3 miligram. Berdasarkan 14 percobaan yang diperoleh dari tabel diatas, nilai rata-rata simpangan atau *error* dari sensor tersebut yaitu sebesar ± 0.15 . Nilai rata-rata simpangan tersebut sangat kecil sehingga sensor *strain gauge* dapat digunakan. Kemudian dibuatlah linieritas hubungan antara kalibrator timbangan digital dengan sensor *strain gauge*.



Gambar 4.8 Grafik Linieritas Sensor *Strain Gauge* Dengan Timbangan Digital

Berdasarkan gambar grafik 4.8 dari tabel 3.11, maka dapat diketahui bahwa hubungan antara timbangan digital dengan sensor *strain gauge* menghasilkan nilai linieritas sebesar $R^2 = 0.9988$. Dengan nilai linieritas pada sensor memiliki nilai yang mendekati 1, maka nilai linieritas yang mendekati 1 tersebut menunjukkan bahwa sensor *strain gauge* yang digunakan sudah bekerja dengan baik dan sesuai dengan nilai hasil pembacaan kalibrator lain.

4.4 Pengujian Lama Waktu Menentukan Berat Barang Berdasarkan Berat atau Volume

Dalam pengujian kali ini, dilakukan pada sistem untuk mengetahui keefektifitasan sistem dari pada harus menggunakan cara manual dalam menentukan berat untuk dieksekusi penentuan harga.

Pada data tabel 3.12, dapat menunjukkan bahwa sistem mampu mengukur berat volume dan berat aktual lebih cepat dari pada menggunakan cara manual. Hal ini dapat menguntungkan karena dapat meminimalisir waktu pengukuran dan penimbangan dengan lebih efektif.

4.5 Pengujian Keberhasilan Sistem

Pengujian sistem secara keseluruhan bertujuan untuk *hardware*, *software* maupun sistem pada alat secara keseluruhan yang dilakukan dengan melihat tingkat keberhasilan sistem. Tingkat keberhasilan sistem didapatkan dari besar nilai rata-rata simpangan atau *error* pada setiap sensor dalam beberapa kali percobaan. Jika nilai *error* yang didapat pada sistem memiliki nilai yang kecil, maka sistem dapat berjalan dengan baik. Berikut tabel data keberhasilan sistem:

Dari tabel pengujian 3.13, maka dapat diketahui dari 8 kali percobaan terhadap sistem mengalami kesalahan atau *error* sebesar ± 0.2 pada pembacaan penimbangan berat aktual dan ± 0.18 pada pembacaan sensor dalam memperoleh panjang, lebar dan tinggi untuk mendapatkan nilai berat volume. Sistem tersebut dalam 8 kali percobaan memiliki nilai *error* yang kecil, sehingga sistem keseluruhan dapat bekerja dengan baik.

Kesalahan terhadap sistem tersebut dapat disebabkan karena kurang tepatnya pembacaan sensor SRF dan sensor *strain gauge* untuk menghitung berat volume dan berat aktual pada barang.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari kegiatan pengujian Tugas Akhir dengan judul “Otomatisasi Penentuan Harga Berdasarkan Berat dan Volume Barang Pada Jasa Pengiriman” yang sudah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Untuk menentukan berat benda pada tugas akhir ini menggunakan sensor *straingauge*, sedangkan volume barang diukur menggunakan sensor SRF-05 melalui pengukuran panjang, lebar, dan tinggi.
2. Hasil perancangan menunjukkan bahwa simpangan rata-rata terhadap pengukuran berat adalah $\pm 0,2$ sedangkan pengukuran volume sebesar $\pm 0,18$.

5.2 Saran

Penulis mengharapkan agar ke depannya alat ini bisa dikembangkan sehingga lebih baik lagi dalam hal penentuan harga pada barang. Beberapa saran yang dapat penulis sampaikan adalah sebagai berikut:

1. Dalam pengukuran nilai volume dan berat diperhitungkan terlebih dahulu agar didapatkan pembacaan hasil yang lebih akurat.
2. Disarankan untuk melakukan pemilihan komponen, alat dan bahan yang tepat untuk membuat otomatisasi penentuan harga berdasar berat dan volume barang, karena kurang tepatnya pemilihan

komponen menyebabkan pengukuran berat dan volume yang terbaca kurang stabil.

3. Kerangka mekanik alat dalam sistem ini diharapkan dapat diperbaiki agar keluaran setiap sensor dapat stabil dan sesuai dengan kalibrator.



DAFTAR PUSTAKA

- Aji, Panji. 2013. "Pemrograman Arduino". [Online] tersedia pada : <http://fandi.students.uui.ac.id/2013/04/30/pemrograman-arduino/>. [Diakses pada: 5 Mei 2016]
- Febrianto, Andri. 2015. "Rancang Bangun Pengereng Kacang Secara Otomatis Dengan Metoda Vakum (Bagian II)". Tugas Akhir D3 Otomasi Sistem Instrumentasi, FVOKASI, Universitas Airlangga, Surabaya.
- Satria, Hadian., Budi, Pono dan Sujono, Mardiyanto. 2007. "Perancangan Dan Implementasi Alat Pengukur Kardus Secara Otomatis". Volume 9. Tesla Jurnal Teknik Elektro, Universitas Tarumanegara, Jakarta.
- Soegijanto, Tono. 2014. *Cara Menghitung Ongkos Kirim Paket*. [Online] tersedia pada : <http://www.kurir.web.id>. [Diakses pada : 27 November 2015]
- Suhendri. 2013. "Sensor Jarak Ultrasonik SRF-05". [Online] tersedia pada : <http://belajar-dasar-pemrograman.blogspot.co.id>. [Diakses pada : 27 November 2015]
- Taufik, Azzi. 2014. "Mikrokontroler Arduino Uno". [Online] tersedia pada : <http://dialogsimponi.blogspot.co.id/2014/11/normal-0-false-false-false-in-x-none-x.html>. [Diakses pada : 5 mei 2016]

LAMPIRAN



24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

FEATURES

- Two selectable differential input channels
- On-chip active low noise PGA with selectable gain of 32, 64 and 128
- On-chip power supply regulator for load-cell and ADC analog power supply
- On-chip oscillator requiring no external component with optional external crystal
- On-chip power-on-reset
- Simple digital control and serial interface: pin-driven controls, no programming needed
- Selectable 10SPS or 80SPS output data rate
- Simultaneous 50 and 60Hz supply rejection
- Current consumption including on-chip analog power supply regulator:
 - normal operation $< 1.5\text{mA}$, power down $< 1\mu\text{A}$
- Operation supply voltage range: $2.6 \sim 5.5\text{V}$
- Operation temperature range: $-40 \sim +85^\circ\text{C}$
- 16 pin SOP-16 package

APPLICATIONS

- Weigh Scales
- Industrial Process Control

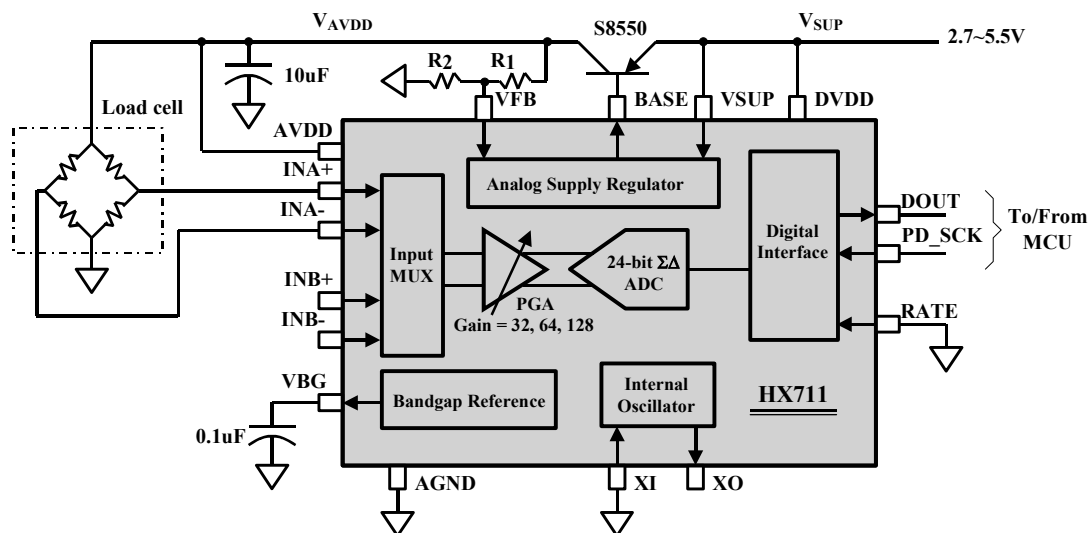


Fig. 1 Typical weigh scale application block diagram

Pin Description

Regulator Power	VSUP	1	16	DVDD	Digital Power
Regulator Control Output	BASE	2	15	RATE	Output Data Rate Control Input
Analog Power	AVDD	3	14	XI	Crystal I/O and External Clock Input
Regulator Control Input	VFB	4	13	XO	Crystal I/O
Analog Ground	AGND	5	12	DOUT	Serial Data Output
Reference Bypass	VBG	6	11	PD_SCK	Power Down and Serial Clock Input
Ch. A Negative Input	INNA	7	10	INPB	Ch. B Positive Input
Ch. A Positive Input	INPA	8	9	INNBNB	Ch. B Negative Input

SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output (NC when not used)
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input (connect to AGND when not used)
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used)
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1 Pin Description

KEY ELECTRICAL CHARACTERISTICS

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(inp)-V(inn)	± 0.5(AVDD/GAIN)			V
Common mode input		AGND+1.2	AVDD-1.3		V
Output data rate	Internal Oscillator, RATE = 0	10			Hz
	Internal Oscillator, RATE = DVDD	80			
	Crystal or external clock, RATE = 0	f _{clk} /1,105,920			
	Crystal or external clock, RATE = DVDD	f _{clk} /138,240			
Output data coding	2’s complement	800000	7FFFFFFF		HEX
Output settling time ⁽¹⁾	RATE = 0	400			ms
	RATE = DVDD	50			
Input offset drift	Gain = 128	0.2			mV
	Gain = 64	0.4			
Input noise	Gain = 128, RATE = 0	50			nV(rms)
	Gain = 128, RATE = DVDD	90			
Temperature drift	Input offset (Gain = 128)	±6			nV/°C
	Gain (Gain = 128)	±5			ppm/°C
Input common mode rejection	Gain = 128, RATE = 0	100			dB
Power supply rejection	Gain = 128, RATE = 0	100			dB
Reference bypass (V _{BG})		1.25			V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6	5.5		V
	AVDD, VSUP	2.6	5.5		
Analog supply current (including regulator)	Normal	1400			μA
	Power down	0.3			
Digital supply current	Normal	100			μA
	Power down	0.2			

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Table 2 Key Electrical Characteristics

Analog Inputs

Channel A differential input is designed to interface directly with a bridge sensor's differential output. It can be programmed with a gain of 128 or 64. The large gains are needed to accommodate the small output signal from the sensor. When 5V supply is used at the AVDD pin, these gains correspond to a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively.

Channel B differential input has a fixed gain of 32. The full-scale input voltage range is $\pm 80\text{mV}$, when 5V supply is used at the AVDD pin.

Power Supply Options

Digital power supply (DVDD) should be the same power supply as the MCU power supply.

When using internal analog supply regulator, the dropout voltage of the regulator depends on the external transistor used. The output voltage is equal to $V_{AVDD} = V_{BG} * (R1 + R2) / R1$ (Fig. 1). This voltage should be designed with a minimum of 100mV below VSUP voltage.

If the on-chip analog supply regulator is not used, the VSUP pin should be connected to either AVDD or DVDD, depending on which voltage is higher. Pin VFB should be connected to Ground and pin BASE becomes NC. The external 0.1 μF bypass capacitor shown on Fig. 1 at the VBG output pin is then not needed.

Clock Source Options

By connecting pin XI to Ground, the on-chip oscillator is activated. The nominal output data rate when using the internal oscillator is 10 (RATE=0) or 80SPS (RATE=1).

If accurate output data rate is needed, crystal or external reference clock can be used. A crystal can be directly connected across XI and XO pins. An external clock can be connected to XI pin, through a 20pF ac coupled capacitor. This external clock is not required to be a square wave. It can come directly from the crystal output pin of the MCU chip, with amplitude as low as 150 mV.

When using a crystal or an external clock, the internal oscillator is automatically powered down.

Output Data Rate and Format

When using the on-chip oscillator, output data rate is typically 10 (RATE=0) or 80SPS (RATE=1).

When using external clock or crystal, output data rate is directly proportional to the clock or crystal frequency. Using 11.0592MHz clock or crystal results in an accurate 10 (RATE=0) or 80SPS (RATE=1) output data rate.

The output 24 bits of data is in 2's complement format. When input differential signal goes out of the 24 bit range, the output data will be saturated at 800000h (MIN) or 7FFFFFFh (MAX), until the input signal comes back to the input range.

Serial Interface

Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls.

When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25th pulse at PD_SCK input will pull DOUT pin back to high (Fig.2).

Input and gain selection is controlled by the number of the input PD_SCK pulses (Table 3). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 3 Input Channel and Gain Selection

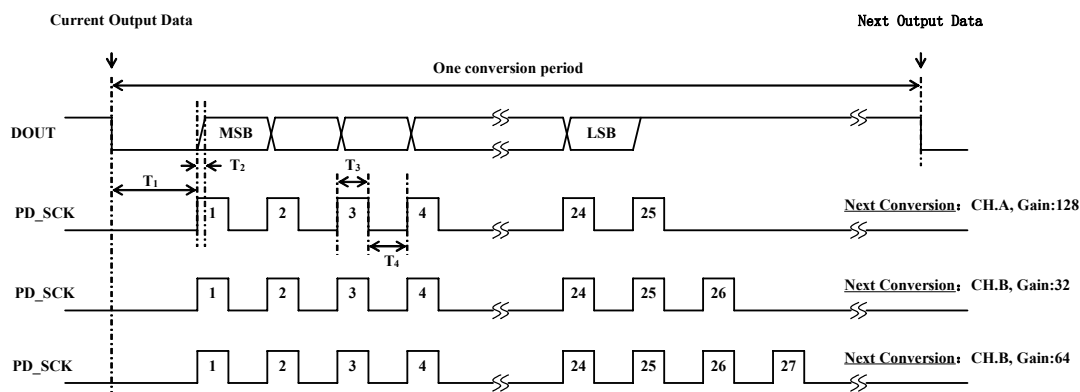


Fig.2 Data output, input and gain selection timing and control

Symbol	Note	MIN	TYP	MAX	Unit
T ₁	DOUT falling edge to PD_SCK rising edge	0.1			μs
T ₂	PD_SCK rising edge to DOUT data ready			0.1	μs
T ₃	PD_SCK high time	0.2	1	50	μs
T ₄	PD_SCK low time	0.2	1		μs

Reset and Power-Down

When chip is powered up, on-chip power on rest circuitry will reset the chip.

Pin PD_SCK input is used to power down the HX711. When PD_SCK Input is low, chip is in normal working mode.

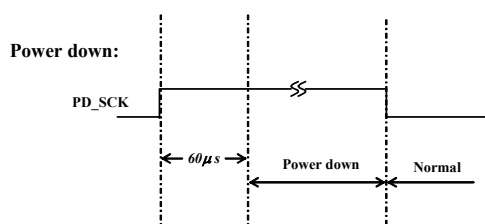


Fig.3 Power down control

When PD_SCK pin changes from low to high and stays at high for longer than 60μs, HX711 enters power down mode (Fig.3). When internal regulator is used for HX711 and the external transducer, both HX711 and the transducer will be

powered down. When PD_SCK returns to low, chip will reset and enter normal operation mode.

After a reset or power-down event, input selection is default to Channel A with a gain of 128.

Application Example

Fig.1 is a typical weigh scale application using HX711. It uses on-chip oscillator (XI=0), 10Hz output data rate (RATE=0). A Single power supply (2.7~5.5V) comes directly from MCU power supply. Channel B can be used for battery level detection. The related circuitry is not shown on Fig. 1.

Reference PCB Board (Single Layer)

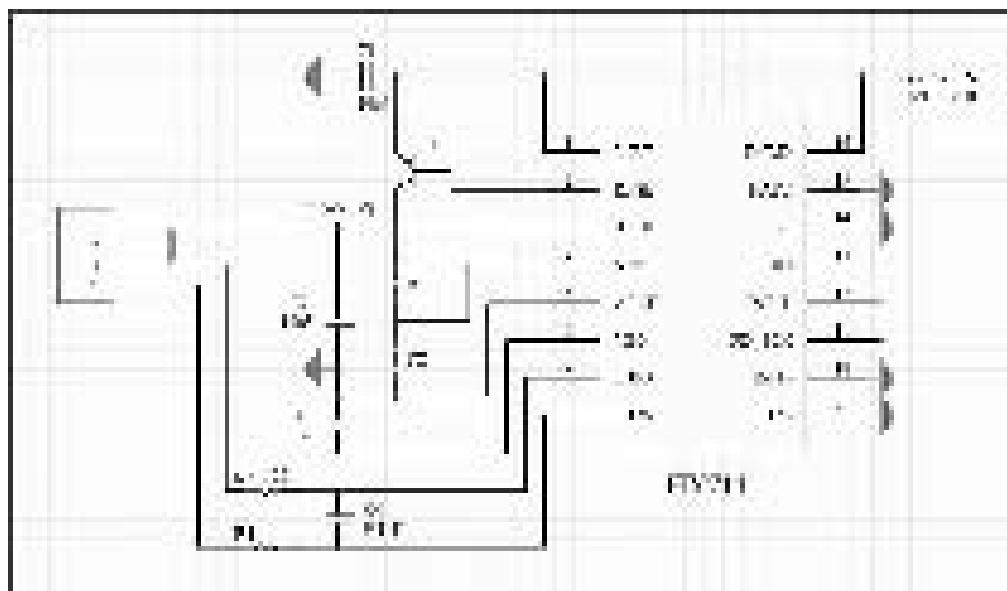


Fig.4 Reference PCB board schematic

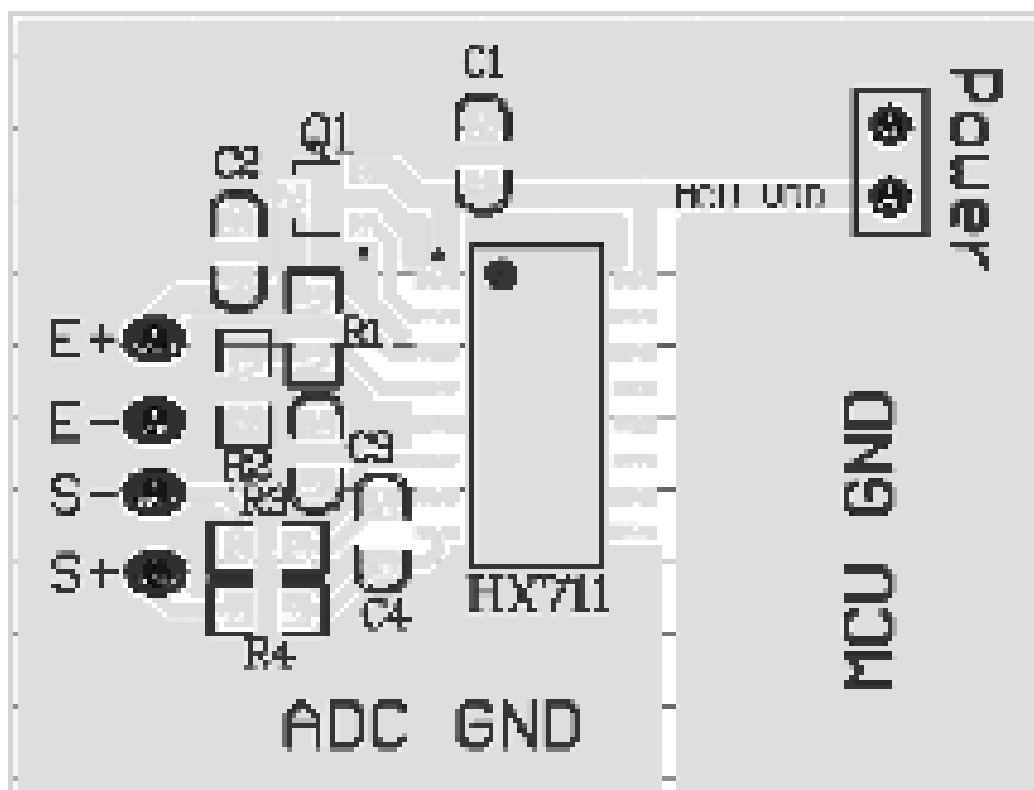


Fig.5 Reference PCB board layout

Reference Driver (Assembly)

```

/*-----
Call from ASM:      LCALL   ReaAD
Call from C:      extern unsigned long ReadAD(void);
                    .
                    .
                    unsigned long data;
                    data=ReadAD();
                    .
                    .
-----*/

PUBLIC      ReadAD
HX711ROM    segment   code
rseg       HX711ROM

sbit       ADD0 = P1.5;
sbit       ADSK = P0.0;
/*-----
OUT:   R4, R5, R6, R7   R7=>LSB
-----*/

ReadAD:
    CLR     ADSK           //AD Enable (PD_SCK set low)
    SETB    ADD0           //Enable 51CPU I/O
    JB      ADD0,$         //AD conversion completed?
    MOV     R4,#24

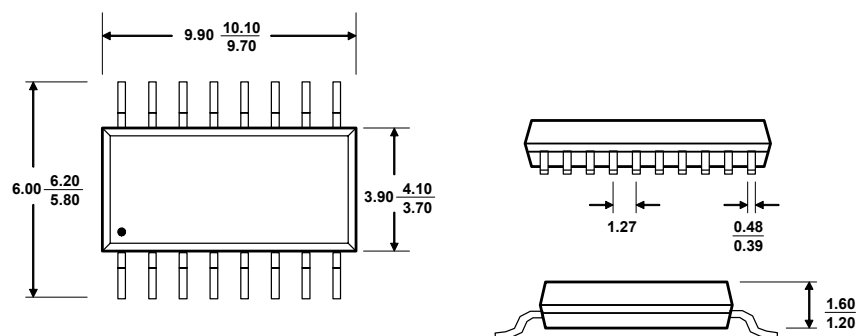
ShiftOut:
    SETB    ADSK           //PD_SCK set high (positive pulse)
    NOP
    CLR     ADSK           //PD_SCK set low
    MOV     C,ADD0         //read on bit
    XCH     A,R7           //move data
    RLC     A
    XCH     A,R7
    XCH     A,R6
    RLC     A
    XCH     A,R6
    XCH     A,R5
    RLC     A
    XCH     A,R5
    DJNZ    R4,ShiftOut    //moved 24BIT?
    SETB    ADSK
    NOP
    CLR     ADSK
    RET
    END

```

Reference Driver (C)

```
//-----  
sbit ADD0 = P1^5;  
sbit ADSK = P0^0;  
unsigned long ReadCount(void) {  
    unsigned long Count;  
    unsigned char i;  
    ADD0=1;  
    ADSK=0;  
    Count=0;  
    while(ADD0);  
    for (i=0;i<24;i++) {  
        ADSK=1;  
        Count=Count<<1;  
        ADSK=0;  
        if(ADD0) Count++;  
    }  
    ADSK=1;  
    Count=Count^0x800000;  
    ADSK=0;  
    return(Count);  
}
```

Package Dimensions



Typ	<u>MAX</u>	Unit: mm
	MIN	

SOP-16L Package

SRF005 ULTRASONIC RANGE SENSOR

Specification:

The SRF005 ultrasonic range sensor detects objects in its path and can be used to calculate the range to the object. It is sensitive enough to detect a 3cm diameter broom handle at a distance of over 3m.

Voltage	- 5V
Current	- 30mA Typ. 50mA Max.
Frequency	- 40KHz
Max Range	- 3 m
Min Range	- 3 cm
Sensitivity	- Detect 3cm diameter broom handle at > 3 m
Input Trigger	- 10uS Min. TTL level pulse
Echo Pulse	- Positive TTL level signal, width proportional to range.
Small Size	- 43mm x 20mm x 17mm height



The module can be used in two different modes:

- Single Pin - Single microcontroller pin (08M, and all M2 and X2 parts)
- Dual Pin - Separate PICAXE microcontroller trigger and echo pins

Most users using the latest generation (M2 and X2) PICAXE parts should select 'single pin' connection mode.

Single Pin Connection Mode:

The PICAXE-08M and all M2/X2 parts have bi-directional pins, so the SRF005 can connect to a single i/o pin.

There are two ways to achieve this connection on the SRF005, via the 5 way header or via the 3 way header. The 3 way header is designed to be compatible with 'servo extension leads' (e.g. part DAG001) so is often the preferred method on new designs. The 5 way header is compatible with older SRF005 modules/PCBs.

Using the 5 way header (note +5V and 0V are marked on the SRF005):

+5V	Connect to 5V
Not used	Do not connect
Signal	Connect directly to the PICAXE pin
Mode	Connect to 0V
0V	Connect to 0V

Using the 3 way header (note SIG and 0V are marked on the SRF005):

Signal (SIG)	Connect directly to the PICAXE pin
+5V	Connect to 5V
0V	Connect to 0V

When using the 3 pin header you MUST also solder a wire link between the mode and 0V on the 5 way header (ie a wire link between pads 4 and 5 on the 5 way header).

Take care not to overheat, and therefore damage, the solder connection pads whilst making connections.

Example PICAXE Program 1:

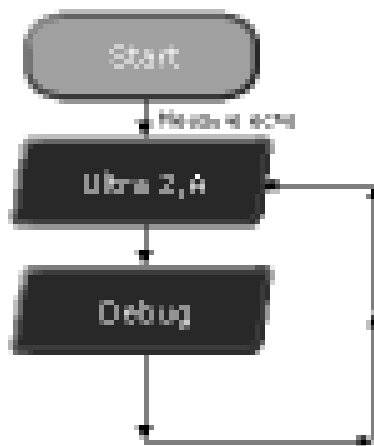
The following program give an example of how to use the SRF005 module with a PICAXE microcontroller in single pin mode. The special 'ultra' command is designed for use with the SRF005 in single pin mode.

```
symbol SIG = C.1          ; Define pin for Trigger & Echo (All M2, X2 parts)
symbol range = w1         ; 16 bit word variable for range

main:
  ultra SIG,range         ; use dedicated ultra command
  debug range             ; display range via debug command
  pause 50               ; short delay
  goto main              ; loop around forever
```

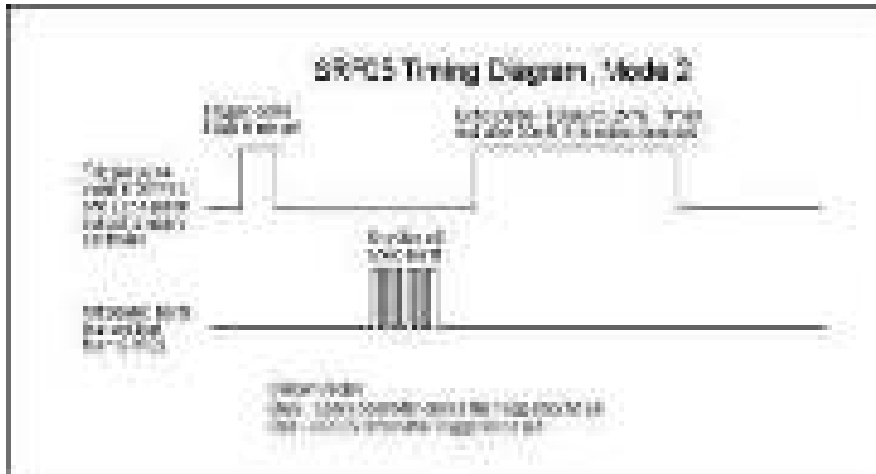
Example Logicator Flowsheet:

The following flowchart give an example of how to use the SRF005 module with a PICAXE microcontroller in single pin mode. The special 'ultra' cell is designed for use with the SRF005.



Technical Details (Single Pin Mode):

The input/Output pin is used to trigger the SRF005 module via a 'pulsout' command and then the pin is converted to an input. The SRF005 module then sends out the sonic burst, and sets the pin high for the time it takes the sonic burst to be returned. Therefore the same PICAXE pin is then used to receive and time this echo pulse via a 'pulsin' command.



The length of the echo pulse is then divided by 5.8 to give a value in cm, and displayed on the computer screen via the 'debug' command. Note that a word variable, w1, is used for the echo timing, as the echo pulse may be a value greater than 255 (maximum value of a byte variable). Word variables are made up of two byte variables and so have a maximum value of 65535 (in this case w1 is made up of b2 and b3, so these two byte variables must not be used anywhere else in the program).

Example Single Pin PICAXE Program 2:

```
symbol SIG = C.1          ; Define pin for Trigger & Echo (All M2, X2 parts)
symbol range = w1         ; 16 bit word variable for range

main:
    pulsout SIG,2          ; produce 20uS trigger pulse (must be minimum of 10uS)
    pulsins SIG,1,range    ; measures the range in 10uS steps
    ; now convert range to cm (divide by 5.8) or inches (divide by 14.8)
    ; as picaxe cannot use 5.8, multiply by 10 then divide by 58 instead
    let range = range * 10 / 58 ; multiply by 10 then divide by 58
    debug range            ; display range via debug command
    pause 50              ; short delay
    goto main             ; and around forever

; Note that X2 parts operate at 8MHz instead of 4MHz and so modify the calculation
; let range = range * 5 / 58 ; multiply by (10/2 = 5) then divide by 58
```

Dual Pin Mode - separate trigger / echo microcontroller pins:

The dual pin mode is used for older PICAXE chips such as the 18X or 28X1.

The SRF005 ultrasonic range finder has 5 connections pins. The 3 pin connector is not used in dual pin mode.

Using the 5 way header (note +5V and 0V are marked on the SRF005):

+5V	Connect to 5V
Echo	Connect directly to PICAXE input pin
Trigger	Connect directly to PICAXE output pin
Mode	Do not connect
0V	Connect to 0V

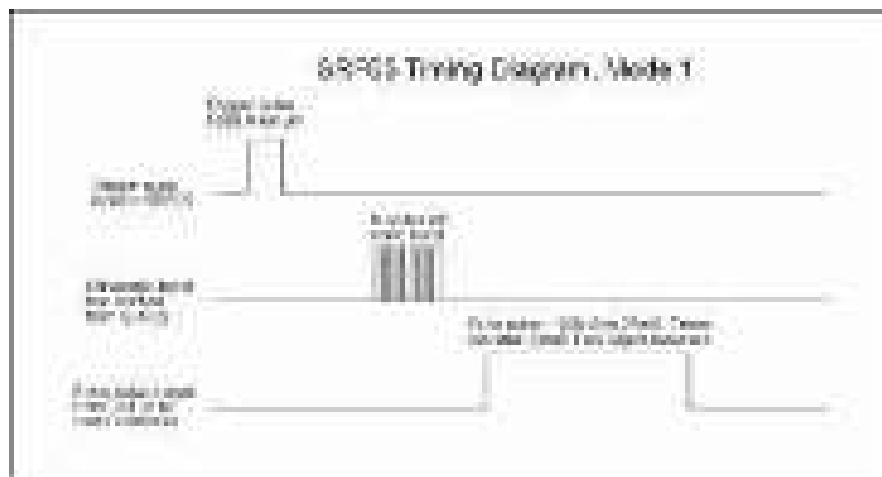
Important - Note that the 'Mode' (pin 4) connection **MUST NOT** be connected for correct operation in this separate trigger/echo mode.

Take care not to overheat, and therefore damage, the solder connection pads whilst making connections.

The SRF005 **Echo Output** is connected to a PICAXE **input** pin.

The SRF005 **Trigger Input** is connected to a PICAXE **output** pin. Note this must be a direct connection to the PICAXE chip leg (do not connect via a darlington driver buffered output on a PICAXE project board).

The following program gives an example of how to use the SRF005 module with a PICAXE microcontroller. Output 3 is used to trigger the SRF005 module via a 'pulsout' command. The SRF005 module then sends out the sonic burst, and sets the Echo Output connection high for the time it takes the sonic burst to be returned. Therefore the PICAXE input (input 6) is used to receive and time this echo pulse via a 'pulsin' command.



The length of the echo pulse is then divided by 5.8 to give a value in cm, and displayed on the computer screen via the 'debug' command. Note that a word variable, w1, is used for the echo timing, as the echo pulse may be a value greater than 255 (maximum value of a byte variable). Word variables are made up of two byte variables and so have a maximum value of 65535 (in this case w1 is made up of b2 and b3, so these two byte variables must not be used anywhere else in the program).

Sample Dual Pin Mode PICAXE Program:

```

symbol trig = 3      ; Define output pin for Trigger pulse (A, M, X, X1 parts)
; symbol trig = b.3   ; Define output pin for Trigger pulse (M2, X2 parts)
symbol echo = 6      ; Define input pin for Echo pulse (A, M, X, X1 parts)
; symbol echo = c.6   ; Define input pin for Echo pulse (M2, X2 parts)
symbol range = w1     ; 16 bit word variable for range

main:
    pulsetrig,2       ; produce 20uS trigger pulse (must be minimum of 10uS)
    pulsinecho,1,range ; measures the range in 10uS steps
    pause 20          ; recharge period after ranging completes
; now convert range to cm (divide by 5.8) or inches (divide by 14.8)
; as picaxe cannot use 5.8, multiply by 10 then divide by 58 instead
    let range = range * 10 / 58      ; multiply by 10 then divide by 58
    debug range                      ; display range via debug command
    goto main                       ; and around forever

; Note that X2 parts operate at 8MHz instead of 4MHz and so modify the calculation
; let range = range * 5 / 58 ; multiply by (10/2 = 5) then divide by 58

```

Example Logicator Flowsheet:

The following flowchart give an example of how to use the SRF005 module with a PICAXE microcontroller in dual pin mode. The special 'ultra' cell is designed for use with the SRF005 and will automatically enable dual pin mode for those PICAXE chips that require it.

